
AMSTRAD
MAGAZINE
SOFTS : 17 LOGICIELS
LISTINGS : 15 PAGES
AMSTRAD DEVIENT "PRO" :
N° 3
18 F
LES DEUX BOMBES
DE LA RENTRÉE
CPC 6128
PCW 8256
Octobre
1985
M 1157 - 03 - 18 F



N° 3
OCTOBRE 1985

SOMMAIRE

News

Softs

Tests machines

Trucs et astuces

Programmation

Listings

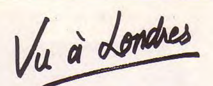
Reportages

Directeur de la publication, rédacteur en chef: Jean Kaminsky. **Coordination de la rédaction:** Philippe Lamigeon. **Rédaction:** Bernard Auré, Michel Chanaud, Eric Charton, Jacques Gilbert, Daniel Martin, Eric Nardeau, Nicolas Seiersen, R.P. Spiegel. **Secrétaire de rédaction:** Mireille Massonnet. **Maquette:** Marc Soria. **Régie publicitaire:** Néo-Média, 55, avenue Jean-Jaurès, 75019 Paris. Tél. 241.81.81. Jean-Yves Primas. **Commission paritaire:** en cours. **Dépôt légal:** 4e trimestre 1985. **Imprimé par** SNIL-RBI. **Edité par:** Laser Magazine, 55, avenue Jean-Jaurès, 75019 Paris.
AMSTRAD MAGAZINE est une publication strictement indépendante et n'a aucun lien vis-à-vis de la société Amstrad.

MEA CULPA

Dans le numéro 2 d'Amstrad Magazine, vous avez certainement tous constaté qu'il manquait une ligne au listing "Garfunkel" page 35 de notre revue. Nous nous excusons de ce contretemps apporté à votre utilisation de ce programme et vous donnons la ligne 770 oubliée (et plus exactement "avalée" par notre imprimante).

```
770 CHAIN MERGE "! garfunkel 2",10000,DELETE 770.
```

LA DMP 2000 : Une nouvelle imprimante pour Amstrad



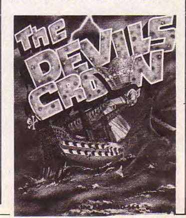
Amstrad présentait une nouvelle imprimante pour ses micros familiaux : la DMP 2000. Elle peut utiliser le feuille à feuille comme le papier continu. La vitesse d'impression peut aller en mode rapide jusqu'à 100 caractères par seconde. La matrice est de 9×9, elle imprime les 96 caractères ASCII plus les accentués internationaux. Plusieurs types de caractères sont possibles : standard, mini, condensé, double standard, double mini, double condensé. La documentation anglaise est bien sûr très prometteuse, il ne nous reste qu'à attendre de voir cette nouvelle machine fonctionner. Elle devrait arriver en France un peut avant Noël.

Les aventures de Bond... Basildon Bond La couronne du diable

Probe Software présentait deux nouveaux logiciels de jeux, dont la seule chose que nous puissions faire à ce jour est d'espérer leur prompte venue en France.

Les aventures de Basildon Bond est un jeu d'arcades, promit comme étant "hilarant"! Basildon Bond, agent secret est chargé d'ailler récupérer Russ Abbot, comédien bien connu en Angleterre, retenu captif dans un donjon, "le plus sombre et le plus profond, là où personne n'ose aller". Brrr... Basildon Bond est le type même de l'anti-héros, il est heureusement secondé par Cooper Man et Blunder Woman. Les trois personnages devront aller sauver l'infortuné captif, mais attention il va falloir penser à tout, même aux choses les plus inimaginables pour parvenir au but.

La couronne du diable est un jeu tout aussi terrible que le



Essayé et testé: le CPC 6128

A peine lancé, le 664 se voit dépassé par son grand frère, le 6128. Reprenant quelques caractéristiques de son prédécesseur, c'est encore une fois une nouvelle machine aux possibilités très intéressantes que nous livre la firme anglaise. Le tout pour un prix typiquement "Amstradien". Merci Monsieur Sugar.

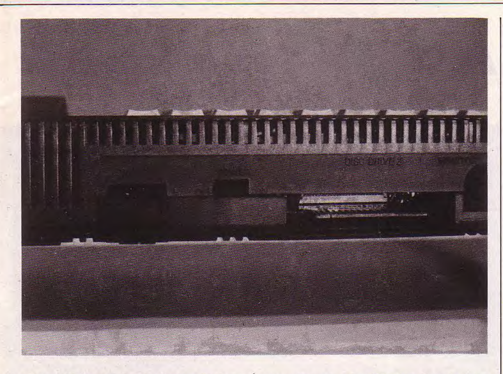
On l'attendait pour le milieu septembre, mais c'est pourtant dès le mois d'août que l'on pouvait découvrir le 6128, exposé chez certains revendeurs, comme "Hyper CB" à Paris.
Il attira bien des curieux le nouveau CPC, l'annonce de la sortie d'un 128 Ko pour un prix si peu élevé doit être pour beaucoup dans ce succès. Il faut bien avouer que l'achat d'un Amstrad actuellement représente un très bon investissement, si l'on faisait un rapport "quantité-qualité-prix".

Tout nouveau, tout beau!

Le moniteur est celui du 664, c'est-à-dire qu'il intégre une sortie 12V pour l'alimentation de l'unité de disquettes. La console est par contre assez différente par rapport à celle du 664, car ses dimensions se sont considérablement réduites. Le clavier est lui aussi différent. Il est d'un aspect beaucoup plus professionnel, si l'on veut faire une certaine comparaison. Les touches au nombre de soixante-quatorze sont de couleur







grise et d'une qualité de frappe équivalente à celle du 664. La première chose que remarquent les habitués des deux précédents CPC est la nouvelle disposition des touches. Le pavé de touches directionnelles et celui des chiffres ne font plus qu'un. La fonction "copy" est venue se positionner à gauche dans le prolongement de la barre d'espacement, tout comme "enter". Pour cette dernière fonction, la touche principale a pris le nom de "return". "Control" est aussi dans le prolongement de la barre d'espacement, mais sa taille est plus importante. Cette nouvelle disposition peut parfois dérouter, surtout au début et si l'on est un habitué des claviers rencontrés précédemment sur les machines de la marque. Le lecteur de disquettes, quant à lui n'a pas bougé et se trouve donc sur le côté droit, pour l'utilisateur.

Pas une extension manquante

Ce n'est pas la réduction assez importante de la console qui a empêché Amstrad d'inclure toutes ses extensions. On retrouve comme sur le 664 les prises et connecteurs suivants: sur le côté gauche, une sortie son stéréo, une prise pour une manette de jeux, une prise DIN pour un lecteur/enregistreur de cassettes. A l'arrière, les connecteurs pour l'imprimante, le second lecteur de disquettes, le bus d'extension, le bouton de mise sous tension et le réglage du volume. On trouve aussi les sorties pour le branchement du moniteur et de l'alimentation 5V et le câble pour l'alimentation de l'unité de disquettes. Le petit haut-parleur est visible dans une partie ajourée, tout comme l'intérieur de la machine grâce aux ouvertures pour les connecteurs. L'utilisation d'un housse de protection semble donc bien utile pendant les périodes où l'on n'utilise pas la machine. C'est aussi valable pour les autres modèles de la marque.

Deux disquettes bien remplies

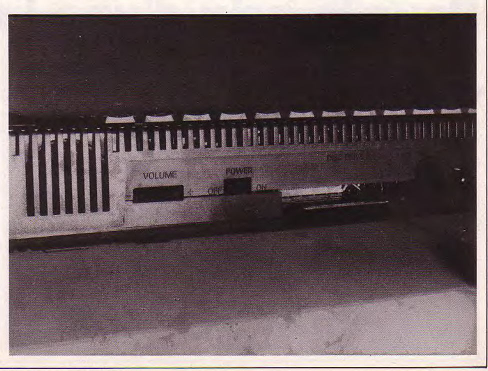
L'éditeur n'a pas changé, mais je reconnais qu'il s'utilise assez facilement. J'avais en effet qualifié d'indigne cet éditeur dans l'article consacré au 664. Ce qui ne m'empêchera pas de dire que je préfère toujours les éditeurs plein écran, avec lesquels on n'utilise pas une touche "copy" pour travailler. A chacun ses goûts, mon cher François!

La machine est fournie avec deux disquettes. Sur la première on trouve le CP/M version 2.2 et DR. LOGO, sur l'autre face un second DR. LOGO et HELP, une aide au programmeur. Sur la deuxième disquette, CP/M Plus c'est-à-dire la version 3.1 et des utilitaires de programmations. Cet ensemble fourni sans supplément de prix est un bel effort du constructeur.

Les manuels livrés avec les machines sont très bien réalisés et celui du 6128 est une mine de renseignements, même s'il est souvent utile de se procurer d'autres ouvrages, ceci pour ceux qui exploitent au maximum leur CPC. Le manuel du 6128 est un gros pavé de 250 pages, soit 500 pages de lectures pour vos nuits blanches. Vous saurez tout sur votre CPC, ses différents systèmes d'exploitations, le Basic en détail, le logo et avec en prime pleins de programmes. Qui dit mieux? Surtout qu'il s'agit du manuel d'origine.

128 Ko, vous êtes vraiment sûr!

La machine à peine déballée, le premier réflexe est de vérifier la présence des 128 Ko de mémoire. L'exécution d'un "Print Fre" ne donnera pour réponse que 42 Ko sous Basic. Où sont donc passés les 64 Ko supplémentaires? La réponse est simple, un processeur de huit bits ne peut adresser que 64 Ko. Donc pour utiliser les 64 Ko supplémentaires on va utiliser un subterfuge simple. Ces kilos — octets supplémentaires vont être divisés en quatre parties (bank) de 16 Ko chacune.



AMSTRAD PCW 8256

Pendant le Personal Computer World Show qui se tenait à Londres du 4 au 8 septembre, nous n'avons pas eu beaucoup de chemin à parcourir pour découvrir le stand Amstrad. Celui-ci se trouvait face à la porte principale d'entrée. Nous nous précipitâmes pour y admirer la dernière machine de la marque, le "Personal Computer Wordprocessor 8256" (8 pour désigner le microprocesseur et 256 pour la capacité mémoire). La présence de cette nouveauté avait été annoncée par notre confrère britannique "Popular Computing Weekly" dans son édition hebdomadaire datée 29 août-4 septembre.



Pour le prix d'une imprimante

Une stratégie déroutante

Cette surprise de dernière minute nous étonna quelque peu, car comme pour le 6128 ce fut au dernier instant que l'annonce officielle tomba. Déjà, la sortie du 6128 prévue à l'origine en France pour le 15 septembre fut effective dès le milieu août, provoqua du même coup une baisse des prix de vente des CPC 464 et 664 ; mais aussi un certain grincement de dents de la part des clients qui avait passé commande d'un 664 et qui se sentaient floués.
Or, que vont-ils penser maintenant avec ce nouveau venu, alors que certains ont acheté leur machine pour effectuer principalement du traitement de texte ?
La stratégie d'Amstrad est parfois déroutante, mais elle prouve au moins qu'avec un bon vieux Z80 on est encore capable de réaliser de belles choses et que, la firme est certainement l'une des plus dynamiques en Europe. D'aucuns avancent également l'étude d'un nouveau produit architecturé autour d'un micro-processeur 16 bits. N'est-ce qu'une rumeur dûe au succès d'Amstrad ? On aura certainement la réponse bientôt.

Une machine professionnelle

Le PCW 8256 a un design très professionnel. Carrosserie de couleur crème pour le moniteur, la console et l'imprimante. Le clavier est différent par rapport à celui du 6128. Il comporte des touches de fonctions supplémentaires, soit au total 82. La console est très plate. Le moniteur haute résolution, offre 32 lignes de 90 caractères en mode texte. Il intègre également une unité de disquettes 3'', située sur la droite de l'écran pour l'utilisateur. L'imprimante est assez compacte et peut recevoir du papier à entête pour les lettres, ou du papier à bande perforée.
Le PCW est disponible en version vert et noir, la couleur n'étant aucunement nécessaire, puisque la machine est avant tout un outil de travail.

Un traitement de texte performant

La machine dispose de 256 Ko de mémoire vive, organisée en ''bank'',



dont 112 Ko sont utilisables comme disque virtuel. Le lecteur de disquette supporte 180 Ko formatés, mais on annonce un lecteur supplémentaire de 1 méga-octets (Mo), donnant environ 720 Ko formatés. Ce dernier devrait être compatible avec les autres lecteurs et donc satisfaire tous ceux qui recherchent un "esclave" de très grande capacité.

Le programme de traitement de texte est très performant, et pourra satisfaire les besoins de la plupart des applications de ce type. Piloté à partir des touches de fonctions du clavier, on voit apparaître des menus dont l'option est sélectionnable à l'aide d'une lettre. Ce qui favorise un travail d'une souplesse extraordinaire.

Durant les démonstrations organisées sur le stand, nous avons pu constater que ces visiteurs n'ayant visiblement pas l'habitude de travailler avec un traitement de texte, et certainement encore moins avec un micro-ordinateur, se familiarisaient avec l'ensemble en l'espace de quelques minutes.

Il est possible de manipuler le texte en création très simplement, grâce notamment au "disque virtuel". De plus on indique toutes les formes de caractères que l'on souhaite obtenir à l'impression durant la saisie, ou la mise en forme du texte.

Une imprimante de bonne qualité

L'imprimante, matricielle à impact, offre différents types de caractères et fonctionne entre 20 coups par secondes (cps), pour la qualité courrier et 90 cps, pour une qualité d'impression standard. Les types de caractères sont : l'élite (standard), le pica, le condensé, le concentré et l'italique. Tous ces modes peuvent être utilisés directement, élargis ou soulignés. Pour la qualité courrier, l'impression se fait en double passe de façon à éliminer le





maximum de défauts. La vitesse utilisée est donc la plus lente. Mais comme il est possible de continuer à travailler un texte pendant qu'un autre s'imprime, on ne ressent pas de gêne véritable.

Il est très désagréable de travailler à proximité d'imprimantes aussi bruyantes qu'un marteau piqueur. Celle fournit par Amstrad a un volume sonore pas trop élevé, n'entraînant pas de perturbations marquantes pour l'utilisateur.

Logiciels et périphériques

Le PCW 8256 est avant tout un ordinateur, de ce fait Amstrad ne l'a pas bloqué en simple machine de traitement de textes. En effet, ce dernier n'est pas intégré en mémoire morte, permettant d'envisager des mises à jour dans le futur. La machine est livrée avec, comme nous a habitué Amstrad, un manuel aussi complet qu'épais. Le Basic est nouveau, il se nomme "Mallard Basic". On trouve également le CP/M plus, le D.R. Logo, et le système graphique GSX de Digital Research. En option on peut citer une interface RS 232 Centronics et série, pour connecter tout ce que l'on souhaite sur le bus d'extension. Côté logiciels, il est possible d'utiliser des programmes professionnels comme Multiplan ou Supercalc.

Le prix du PCW 8256 tournerait aux alentours de 7 000 F. Car le tout est livré en un seul lot, console-moniteur-imprimante. De quoi attaquer les applications professionnelles pour le prix d'une bonne imprimante. Il va en faire des heureux Mister Sugar !

Philippe Lamigeon



premier. Imaginez plutôt : dans les temps anciens, un navire pirate, chargé de plein de trésors, sombra à la suite d'une rencontre, lors d'une nuit de tempête, avec un immense diable couronné. Aujourd'hui, il s'agit d'oser plonger pour aller récupérer les nombreux trésors enfouis sous les mers, la couronne du diable (tombée dans le combat). Le tout au milieu de poissons tueurs et autres douceurs. Et avec tout çà, il faut en plus éviter de mourir !

Faites-le comme des professionnels

Macmillan Software présentait à Londres trois nouveaux jeux pour CPC 464 et 664 (entre autres micros-ordinateurs). Des jeux probablement considérés comme géniaux puisque l'affiche alléchante était "Faites-le comme des professionnels".
Concrètement, il s'agit d'être des "pros" du football, de la magie et du cinéma. World Cup Soccer est partagé en deux : World Cup Manger où vous constituez votre club de foot et World Cup Factfile où vous apprenez tout sur toutes les coupes du monde : le nom et les performances des joueurs, les résultas des matchs...
Magic vous montre et vous fait comprendre une quinzaine de tours fantastiques. Vous pouvez alors réaliser votre propre "show".

Enfin Screenplay vous transforme en metteur en scène : créez vos propres personnages, inventez vos dialogues, vos décors... Les trois jeux sont commercialisés en Grande-Bretagne à 8,95 livres ce qui correspondrait environ en France à 110 F.



Azimuth 2000 : tête d'alignement pour votre magnéto à cassette

Interceptor Ltd montrait un astucieux "Azimuth 2000" permettant d'aligner exactement votre bande magnétique avec la tête de lecture du magnétophone à cassette. Cela permet entre autre de charger exactement le programme ou le bout de programme souhaité. Le "package" comprenant l'Azimuth 2000 lui-même, un jeu de test "Chopper squad", un manuel d'instructions, ne fonctionne que pour le CPC 464.

En direct des clubs

Micro-soft Informatique: des stages pour vous

L'association Micro-soft Informatique* est amoureuse, c'est elle-même qui le dit, des micros-ordinateurs Amstrad. En conséquence de quoi, elle fait la formation des futurs micro-programmeurs de génie sur du matériel Amstrad, et uniquement du matériel Amstrad. Ces stages, aussi bien d'initiation que de perfectionnement, sont "offerts" et essaient d'être le plus adaptés aux besoins spécifiques des différents utilisateurs.

** Renseignements et inscriptions: M.S.I., 1 bis rue Riauz 33800 Bordeaux Tél.: (56) 94.50.94.*

Amstrad... Gram: un nouveau bulletin des utilisateurs

Amsclub-France: devinez un peu ce que çà peut bien être? Hein? On vous le demande? ... Allez, on est sympas, on vous le dit: un nouveau club d'utilisateurs de de l'Amstrad. Non, pardon: une nouvelle association, loi 1901. Et Amstrad... Gram (il fallait bien la faire celle-là!), c'est leur bulletin.
Amsclub est créé depuis mai 1985, le numéro 1 d'Amstrad Gram est daté de juin, les numéros suivants devraient se succéder tous les deux ou trois mois. Attention! Amsclub France ne veut pas être une association parmi beaucoup d'autres sans spécificités particulières. Son ambition est "noble" aider les utilisateurs à dépasser le strict niveau du programmeur autodidacte, souvent trop vite limité, selon Jean Le Flour, le vice-président. Pour ce faire, le bulletin de liaison a un rôle important; mais aussi et surtout les réunions mensuelles et les groupes de travail sur des sujets précis. Pour obtenir des renseignements supplémentaires, ou pour adhérer directement (50 F. + 180 F. de cotisation annuelle écrivez à:

Amsclub France, 4 impasse Truillot, 75011 Paris.

Amster vous informe...

Créé en mai, AMSTER est en passe de devenir le club "number one" à l'attention des passionnés de l'AMSTRAD 464/664/6128. Ce succès s'explique par le nombre des activités qu'il propose:
— Permanence technique téléphonique tous les samedi au (1) 869.85.68 (à partir du 24 octobre, composer le (1) 48.69.85.68).
— Gazette bimestrielle avec des programmes inédits, des articles d'initiation, des bancs d'essai de logiciels et de matériels et plein de "trucs et d'astuces" utiles...
— Une coopérative de logiciels et de matériels à des prix défiants toute concurrence...
— Une émission de radio coproduite avec AMSTRAD MAGAZINE et LOGI'STICK, dont voici en exclusivité le sommaire complet:

AMSTER RADIO

— Invitée: Marion Vannier Présidente d'AMSTRAD FRANCE.
— Les nouveautés logiciels et matériels.
— Emission d'un programme par voie hertzienne (grande première).
— Compte rendu de nos découvertes et de nos derniers contacts glanés aux Salons ou AMSTER. C'est rendu, soit: PERSONNAL COMPUTER SHOW et AMSTRAD USER SHOW à LONDRES.
Cette émission animée par AMSTER aura lieu le 11/10/85 de 19 h à 20 h RADIO VOLTAGE 98 MHZ la radio du 93.
Pour ceux qui n'auront pas la chance de nous écouter, il pourront tout de même se procurer l'enregistrement complet de l'émission en s'adressant à AMSTER.
Dès maintenant, inscrivez-vous sans attendre (190 F par an) en écrivant à: AMSTER CLUB, 68 Av. Paul Vaillant couturier, 93120 La Courneuve.
Ou en téléphonant au: 869.85.68

A bientôt, la présidente, V. Petit



Amstrad premiers programmes

Éditeur : *Sybex*
Auteur : *Rodnay Zaks*
Prix : *98 F*

Pour ceux qui connaissent, ce livre est la version Amstrad du premier ouvrage de Rodnay Zacks *"Votre premier programme Basic"*, paru également en France aux éditions Sybex. Cela dit, comme c'est un excellent ouvrage, la ré-édition Amstrad n'est nullement à négliger. Il faut rappeler que Rodnay Zaks est docteur en informatique de l'université de Berkeley en Californie, (donc proche de nombreuses sources), et l'un des premiers enseignants de micro-informatique. Cela se sent dans ses écrits puisqu'il est, à mon avis bien sûr, l'une des rares personnes à réussir à transmettre un savoir scientifique d'une façon sympathique : claire, drôle tout en étant sérieuse, simple.
Mais revenons-en au livre : il se promet de vous apprendre le Basic en une heure. Si, comme Woody Allen vous lisez "Guerre et Paix" en vingt minutes, çà va, sinon c'est peut-être un peu juste ! En tous les cas, ce livre est clair, bourré de diagrammes et d'illustrations en couleur (et marrantes en plus, siouplait), logique. Vous allez apprendre à communiquer avec votre ordinateur, parler Basic, calculer en Basic, mémoriser des valeurs et utiliser des variables, écrire un programme clair... Bref, vous allez apprendre le Basic, du moins vous allez acquérir de solides bases sans vous arracher trop de cheveux, avec le sourire même, de temps en temps.
Comme le dit Rodnay Zaks lui-même, ce livre *"s'adresse aux nouveaux utilisateurs âgés de 7 à 77 ans"*; plus : il leur est indispensable. Faites-en l'acquisition et prouvez donc un peu que les vrais bons autodidactes, çà existe !

Sybex, 6/8, Impasse du Curé, 75008 PARIS.

Le tour de l'Amstrad

Éditeur : *Cedic/Nathan*
Auteurs : *Pierre Raguenes - Gérard Sitbon*
Prix : *80 F*

Ce livre s'adresse à tous les utilisateurs de l'Amstrad, CPC 464 ou CPC 664. Précisons qu'il vaut mieux avoir déjà quelques notions de Basic et quelques idées sur le fonctionnement de son Amstrad pour s'en servir, dans la mesure où le livre s'intéresse avant tout aux nouveautés introduites par Amstrad dans le monde de la micro familiale. Les notions de base sont considérées comme un pré-requis.

Les auteurs, visiblement admiratifs devant leur machine, affirment qu'avec ses possibilités de redéfinition de caractères, de fenêtrage des textes, d'interruption logicielle, de gestion de la musique en temps réel et de redéfinition du clavier, l'Amstrad fait entrer *"la micro-informatique familiale et semi-professionnelle dans l'ère de l'informatique tout court"*. La première partie du livre est donc consacrée à l'étude des instructions du Basic dans les domaines du graphisme, du son, des interruptions et du clavier. La seconde partie s'intéresse de plus en plus près à la machine elle-même : organisation de la mémoire, initiation au langage machine, liste de sous-programmes "machines". Enfin une troisième partie offre des jeux ou utilitaires spécifiques à l'Amstrad.
Un "bouquin" très bien fait, très intéressant pour ceux qui veulent comprendre, maîtriser et améliorer leur "art" de la programmation.

Cedic/Nathan, 6/10, Bd Jourdan, 75014 PARIS.



3D STUNT RIDER



Éditeur : DJL
Distributeur : Amsoft
Support : cassette
Genre : arcade/réflexe
Graphisme : ★★★
Intérêt : ★★
Difficulté : ★★★★
Appréciation : ★★★

Le but du jeu est simple : sauter à moto au-dessus d'une succession de bus (anglais, of course !). L'écran vous montre le guidon de votre moto, ainsi que la visualisation de votre position de conduite, une montre vous indiquant le temps que vous pouvez utiliser pour vous concentrer et enfin, un compteur de vitesse. Au départ vous avez cinq essais possibles, mais ils ne se renouvellent pas à chaque réussite. Il est donc impératif de réussir les sauts à la première tentative. Dès votre accélération, il faut essayer de garder l'axe du tremplin, chose assez difficile dans les niveaux supérieurs. Le vol et l'atterrissage sont très délicats, car la moto (et le pilote) doivent avoir une position parfaite, sinon c'est la chute garantie. A chaque saut, le nombre de bus augmente de 5. Vous commencez avec 25 bus, au niveau 1. Jusqu'au niveau 8, c'est franchissable sans trop de problèmes, mais après cela devient franchement très difficile. Le jeu n'est pas nouveau, mais il est bien réalisé. Le seul défaut est un avion franchement "casse-pieds", qui passe avant chaque tentative. Qui l'a abattu ?

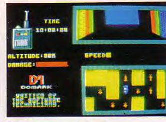
MICRO SAPIEN



Éditeur : Ere informatique
Distributeur : Ere informatique
Support : cassette
Genre : arcade/réflexion
Graphisme : ★
Intérêt : ★
Difficulté : ★★

Pour une surprise, ce fut une belle surprise. C'est le genre de "chose" que l'on n'aimerait plus voir chez un éditeur français. Même si le logiciel répondait à une demande, il valait mieux ce jour-là prendre une journée de congé. Pendant le chargement du logiciel on attend avec impatience car Michel Rho, le graphiste de Ere, nous offre une image de représentation superbe. Mais dès le lancement tout se gâte. Il faut réunir des lettres, alors que l'on est poursuivi par un monstre. Le problème c'est que tout est monstrueux. Il est bien difficile de s'y retrouver. Après cette course poursuite "délirante", il faut composer un mot utilisant le maximum de lettres récoltées. Les points sont attribués ensuite. Un faux pas de l'éditeur parisien à oublier (très) rapidement.

A VIEW TO A KILL



Éditeur : Domark
Distributeur : Eureka
Support : cassette
Genre : aventure/action
Graphisme : ★★★
Intérêt : ★★★★
Difficulté : ★★★★★
Appréciation : ★★★★

Tiré des aventures du dernier James Bond, ce jeu va très certainement devenir un best dans sa catégorie. La présentation, nous offre en prime la musique du groupe britannique "Duran Duran". Là, Domark nous en donne pour notre argent, puisque nous avons trois jeux pour le même prix. Le choix de l'épreuve se fait au début, car il s'agit bien de trois enregistrements différents, et non d'un jeu unique. Vous devrez découvrir une bombe se trouvant dans une mine (Silicon valley mine). Vous pourrez sauter pour éviter les cheminées, ou bien les utiliser pour changer de niveau. Attention, car dans certains cas vous resterez coincé sans possibilité



de vous en sortir. Il faudra pourtant agir vite car la bombe peut exploser, si vous êtes un Bond de troisième catégorie. L'autre épreuve consiste à retrouver Stacey et bien sûr à délivrer la belle. L'aventure se déroule dans les 75 pièces de l'Hôtel de Ville. Il faut utiliser les objets rencontrés. Attention au temps, sinon adieu Stancey. Enfin, le dernier jeu est une course en voiture dans les rues de Paris. La police aux trousses, des conducteurs plus dangereux les uns que les autres, de quoi finir cardiaque. A View to a Kill est vraiment une superbe aventure. Dommage que les graphismes ne soient pas meilleurs.

THE RING OF DARKNESS



Éditeur : Wintersoft
Support : cassette
Genre : jeu de rôle
Graphisme : ★★★
Intérêt : ★★★★
Difficulté : ★★★★
Appréciation : ★★★★

Un véritable jeu de rôle animé et graphique sur AMSTRAD. Le chargement se déroule en deux phases, après un cours moment un menu s'affiche et vous propose les différents critères qui peuvent composer votre personnage (elfe, sorcier, humain, etc.).
Une fois ces données entrées, la partie jeu commence à être chargée. Ici on est un peu dérouté, les graphismes se limitent à un personnage à déplacer dans un univers très stylisé composé de rivières, de montagnes et de forêts.
Ne vous y fiez pas, ce jeu révèle une incroyable complexité, et une grande variété de situations. Tout d'abord vous pouvez déplacer votre personnage au joystick, et intervenir par de courtes phrases tapées au clavier.
De plus les obstacles sont nombreux, guerriers acharnés, donjons infestés de pièges et bien d'autres surprises. Un bon jeu d'aventure malheureusement en anglais.

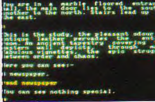
ALIEN 8



Éditeur : Ultimate
Distributeur : Run informatique
Support : cassette
Genre : aventure/action
Graphisme : ★★★★
Intérêt : ★★★★
Difficulté : ★★★★★
Appréciation : ★★★★

Connaissez-vous Night Lore, ce soft d'Ultimate absolument génial ? Non, alors tant pis pour vous. Au début on pensait que l'éditeur anglais avait modifier Night Lore pour nous offrir cet Alien. S'il est vrai qu'il y a des similitudes, on vous garanti que ce jeu est différent du précédent, car en utilisant nos connaissances acquises, nous nous sommes lamentablement "plantées". Alien est un robot qu'il faut guider dans des salles, où l'attendent des épreuves difficiles. Il faut utiliser toute sa ruse et son adresse pour passer à la pièce suivante. Rien n'est épargné au pauvre Alien, par conséquent, à vous-même. Les nerfs craquent rapidement, la folie vous guette, quand à la "tremblote" ne vous inquiétez surtout pas, elle vous gagnera dès le début du jeu. Le jeu est vraiment beau et il serait à notre humble avis dommage de ne pas le posséder. C'est exactement le type de jeu dont on ne se lasse pas facilement, si l'on est bien évidemment amateur du genre. A mettre de côté pour les longues soirées d'hiver.

MORDON'S QUEST



Éditeur : Melbourne house
Support : cassette
Genre : aventure textuelle
Intérêt : ★★★
Difficulté : ★★★★
Appréciation : ★★★

"Vous êtes dans la chambre principale, une atmosphère indéfinissable emplit la pièce comme si une grande tragédie avait eu lieu ici." Telle est la première phase de Mordon's, jeu d'aventure qui prend place dans quelques milliers d'années alors que l'univers au bord de la destruction cherche désespérément un héros pour le sortir de ce mauvais pas.
Ce héros, vous, va devoir au sein d'un univers futuriste, résoudre une aventure extrêmement complexe. Ne cherchez pas les graphismes dans ce jeu : il n'y en a pas, mis à part une superbe page de présentation lors du chargement du jeu. Les auteurs ont préféré réserver toute la mémoire disponible pour l'analyse syntaxique et le texte. Il y ont d'ailleurs fort bien réussi la description des salles étant relativement précise et le temps de réponse aux questions quasiment inéxistant.
Les phrases sont entrées entièrement, le programme de compréhension de celles-ci étant particulièrement sophistiqué. Un dictionnaire de plus de 400 mots permet d'éviter au maximum les "pardons" ou autres "je n'ai pas compris" si fréquemment rencontrés dans les jeux de ce style.



PLANETE BASE



Éditeur : *Loriciels*
Distributeur : *Loriciels*
Support : *cassette*
Genre : *action/éducatif*
Graphisme : ★★★★
Intérêt : ★★★★
Difficulté : ★
Appréciation : ★★★★

Planète base est un jeu d'actions (pas franchement violent) et éducatif à la fois. Il réussit en tout cas très bien sa mission, car il ne manque pas d'intérêt même pour les plus grands. Le but du jeu est de nourrir un gros animal, qui ressemble beaucoup à "Casimir", avec des pommes. Seulement la brave bête ne se nourrit que de pommes rouges. Or sur les arbres on ne trouve que celles de couleurs blan-



che, beige, bleu clair et marine. Il faut donc organiser un échange avec un marchand suivant une base établie au début du jeu. Si par exemple, on cueille 16 pommes blanches on en obtiendra 8 beiges, puis 4 bleues claires, ensuite deux marines pour enfin en gagner une rouge, qui permettra de nourrir l'étrange animal. Si votre base d'échange est plus élevée, vous trouverez sur les arbres les couleurs des pommes autorisant en fin de compte la possibilité d'échange. La cueillette est chronométrée et signalée par le passage d'une planète, de la gauche vers la droite de l'écran. Ce programme ne manque ni de qualité ni de charme. Brâvo à Loriciels de nous faire aimer, pour une fois, un éducatif.

THE SCOUT STEPS OUT



Éditeur : *H.R. software*
Distributeur : *Amsoft*
Support : *cassette*
Genre : *arcade/réflexe*
Graphisme : ★★★
Intérêt : ★★★
Difficulté : ★★★★★
Appréciation : ★★★★

Des grasphismes assez réussis, une bonne douzaine de tableaux, beaucoup de difficultés, une musique drôle : la recette d'un bon jeu. Dans ce jeu, vous devez être l'exemple de votre troupe. Réunis devant le campement, tous vos camarades vous regardent partir pour réaliser vos bonnes actions (les B.A pour les anciens). Vous devez laver les vitres d'un pensionnat, ramasser des champignons en évitant ceux qui sont empoisonnés, récolter des échantillons de la vie marine afin de les étudier par la suite en classe, maîtriser l'électronique pour être capable de réparer la radio du camp. Et ce n'est pas tout, vous aurez d'autres missions. Mais attention, ne croyez surtout pas que l'on vous laissera agir tranquillement. D'autres ont trop d'intérêt à savourer votre échec. Il sera nécessaire de contourner les obstacles en vous servant de votre adresse, mais aussi de votre tête. On aimerait voir plus de jeux de cette qualité sur le catalogue Amsoft.

RED ARROWS



Éditeur : *Database software*
Distributeur : *Run informatique*
Support : *cassette*
Genre : *simulation/pilotage*
Graphisme : ★★★
Intérêt : ★★★
Difficulté : ★★★★★
Appréciation : ★★★

La jaquette de la boîte précise qu'il s'agit du meilleur simulateur jamais écrit pour un ordinateur domestique. Il est vrai qu'il est beau, mais nous sommes restés sur notre faim, car il manque ce petit quelque chose qui en aurait fait un logiciel de très grande qualité. En vol d'entraînement, on ne voit pas grand-chose, mis à part le sol représenté par une bande de couleur verte. Il fut monter relativement haut pour réussir ses premiers loopings, sinon "bonjour" le sol. En vol d'acrobatie, les Red Arrows sont l'équivalent anglais de la "Patrouille de France".
Plusieurs figures doivent être exécutées toujours en formation. Votre place dans le groupe est déterminée, vous devrez donc la respecter. L'image devient plus intéressante, car on voit les avions voisins. Votre leader, le chef de vol, vous indique les modifications à apporter si vous n'êtes pas des plus faciles. C'est précisement le plus grand intérêt de ce logiciel : sa difficulté. Sur le tableau de bord, vous pouvez visualiser la figure à exécuter et votre propre mouvement, ainsi que la position de votre appareil et différents indicateurs dont celui de puissance des réacteurs. On peut regretter que le graphisme ne soit pas plus poussé. Le manuel livré avec le soft vous apprendra beaucoup de choses sur l'histoire des Red Arrows, qui ont eux-mêmes participé à la conception de ce programme. Bon vol.

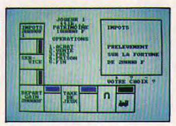
WORLD WAR 3



Éditeur : *Free game blot*
Support : *cassette*
Genre : *wargames*
Graphisme : ★
Intérêt : ★
Difficulté : ★
Appréciation : ★

Un wargame a trois niveaux de jeu simulant un combat entre deux bataillons composés de blindés, etc. Le champ de bataille est représenté par une grille de 5 par 9 secteurs. Les forces sont disposées aléatoirement sur le lieu de la bataille. Vous ne pouvez pas voir vos ennemis, il vous faut donc envoyer vos éclaireurs afin d'identifier leur position. Ce jeu souffre malheureusement d'un important manque de graphisme et les bruitages sont quasiment inexistants. Le manque d'options disponibles rend les parties vites lassantes.

MONOPOLIC



Éditeur : *Free game blot*
Support : *cassette*
Genre : *simulation économique*
Graphisme : ★★
Intérêt : ★★
Difficulté : ★★★
Appréciation : ★★

Le grand classique du jeu de société est traité cette fois par Free game blot. Ce jeu, même s'il souffre d'un grand manque d'originalité n'est pas inintéressant. Le logiciel peut gérer jusqu'à quatre joueurs simultanément, chaque joueur pouvant être représenté par une personne, ou pris en main par l'AMSTRAD (il est ainsi possible de faire jouer l'ordinateur contre lui-même, ou bien d'opposer plusieurs participants à une personne jouant seule). Les principales règles du jeu de société sont fidèlement respectées, le parcours du PDG en herbe étant bien entendu semé d'embûches telles que des taxes diverses, la prison, les cartes de chance qui n'en sont pas toujours, ou bien les loyers qui vident progressivement votre compte en banque.
Les parties sont en général assez longues mais le mouvement devient vite répétitif, voire ennuyeux lorsque l'on affronte l'ordinateur.

RALLY II



Éditeur : *Loriciels*
Distributeur : *Loriciels/Amsoft*
Support : *cassette*
Genre : *course auto*
Graphisme : ★★★
Intérêt : ★★
Difficulté : ★★
Appréciation : ★★
Prix indicatif : *160 F*

Si vous êtes un passionné de la course automobile, ce jeu devrait vous intéresser. Vous devez piloter une voiture qui participe à un rallye, on s'en serait douté, se disputant sur dix étapes. Ainsi vous affronterez le sable, la neige, le brouillard, la nuit, la course sur pont, sur bord de mer ou encore sur une route ensoleillée. Le gros problème de ce Loriciels, c'est d'être d'une monotonie effrayante. Sur les parties difficiles, mise à part l'épreuve sur pont qui peut poser des problèmes, toutes les autres étapes se passent sans grandes difficultés. De plus, la neige ou le sable ne sont pas des handicaps. Avec un peu de malice, vous pouvez terminer toutes les épreuves en passant sur les côtés, évitant ainsi les concurrentes que vous pourriez percuter. Ceci fait que le niveau de difficulté du jeu est vraiment ridicule, permettant d'accéder au niveau supérieur sans trop de problèmes. Encore un reproche, toutes les parties du rallye ont le même circuit. Il est pourtant possible de modifier le programme initial, en augmentant par exemple la valeur d'une courbe, en changeant des portions de circuit. Malgré cela, il est toujours aussi facile de passer sur le côté. Vraiment pas génial ce programme.



Récupérez vos programmes ou fichiers protégés

Vous avez écrit un programme, puis l'avez enregistré sur une cassette ou une disquette en le protégeant. Après quoi, vous l'avez effacé de la mémoire de votre AMSTRAD. Malheureusement, dans votre précipitation de mettre à l'abri votre œuvre, vous avez oublié de conserver une copie non protégée de votre travail.

Le programme proposé ici vous permettra de recopier vos fichiers de cassettes, ou disquette à cassette, ou disquette en supprimant le flag de protection mis en place par le soft du CPC. Il vous est fourni sous la forme d'un listing Assembleur et d'un listing Basic. Le listing Assembleur permettra à ceux qui le désirent d'améliorer ce programme. (Par exemple en y incluant un traitement des erreurs). Le listing Basic autorise tous les possesseurs de CPC à utiliser le programme et permet de choisir le type de support des fichiers entrée et sortie.

Programme Assembleur

Le programme Assembleur qui vous est présenté aurait pu être écrit par tout possesseur de la précieuse brochure *AMSTRAD COMPLETE FIRMWARE SPECIFICATIONS."*

Il permet, pour tout fichier enregistré en *"FILE FORMAT"* (c'est le cas des fichiers ou programme créés par la commande SAVE) de charger ce fichier en RAM, puis de la ré-enregistrer sur un support magnétique. Le fichier obtenu en sortie aura des spécifications strictement identiques au fichier fourni en entrée, mais il ne sera plus protégé. Cela signifie que, à l'exception du bit de protection, les différentes parties, utiles du header du fichier entrée (adresse des données, adresse du point d'entrée, longueur du fichier, zone utilisateur) auront les mêmes valeurs dans le header du fichier sortie.

La routine AFFER1 est destinée à permettre le traitement des erreurs. Elle n'a pas été utilisée dans ce programme et se compose uniquement de deux instructions, mais vous pouvez éventuellement y inclure le traitement d'erreurs que vous désirez. Attention toutefois, dans ce cas, à modifier l'adresse de chargement du programme. Celle qui est donnée (&A4B5) a été calculée pour permettre aux possesseurs d'une unité de disquettes d'utiliser ce logiciel.

Programme Basic

Le programme BASIC permettra à ceux qui ne disposent pas d'un Assembleur-Éditeur de mettre en œuvre ce programme. Il permet en outre à ceux qui disposent d'une unité de disquettes de choisir le support du fichier entrée et celui du fichier sortie. Pour pouvoir utiliser ce programme, les utilisateurs AMSTRAD qui ne possèdent pas d'unité de disquettes ne devront pas saisir les lignes 50, 60, 70, 100, 110, 130 et 140.

Données et routines utilisées en Assembleur

TYP-IN type de support du fichier entrée. sera à 0 pour une cassette, à 1 pour une disquette.
TY-OU type de support du fichier en sortie. sera à 0 pour une cassette, à 1 pour une disquette. Ces deux zones sont renseignées par le programme Basic. Elles permettent de déterminer la longueur maximum du nom de fichier.
— NOM-IN nom du fichier en entrée.
— NOM-OU nom du fichier en sortie.
— LNG-IN longueur du nom du fichier entrée.
— LNG-OU longueur du nom du fichier sortie.
— TYP-PG type du fichier entrée.
— ADR-DE adresse de chargement du fichier.
— ADR-EN adresse du point d'entrée.
— LNG-PG longueur totale du fichier.
— USR-ZO zone utilisateur du header.
Les zones TYP-PG, ADR-DE, ADR-EN, LNG-PG, USR-ZO, sont renseignées par le heuder du fichier entrée.
Les zones NOM-IN, NOM-OU, LNG-IN, LNG-OU sont renseignées par la demande de nom de fichier à l'écran.

Principales routines utilisées

Ces routines sont appelées par un CALL à leur adresse. Les adresses d'appel sont les suivantes:
BC77 OPEN d'un fichier input. Cette routine retourne:
+ Dans le registre A le type du fichier,
+ Dans le double registre BC la longueur totale du fichier,
+ Dans le double registre DE l'adresse de chargement des données,
+ Dans le double registre HL l'adresse d'un buffer contenant le header du fichier.
La zone utilisateur est contenue dans les octets 28 à 63 de ce buffer.
+ BC83 READ d'un fichier input. Cette routine retourne dans le double registre HL l'adresse du point d'entrée.
BC8C OPEN d'un fichier sortie. Cette routine retourne dans le double registre HL l'adresse d'un buffer contenant le header. La zone utilisateur obtenue lors de l'OPEN du fichier entrée sera reconduite dans ce buffer.
BC98 WRITE d'un fichier out put.
BC0E Mise de l'écran dans le mode fourni dans le registre A.
BC32 affectation à l'INK dont le numéro est donné dans le registre A des couleurs dont les numéros sont donés dans les registres B et C.
BB06 Lecture d'un caractère frappé au clavier. Le caractère est donné dans le registre A.
BB5A Affichage à l'écran du caractère dont la valeur est dans le registre A.
BB78 Retourne les positions horizontales du curseur dans les registres H et L.
BB75 positionne le curseur aux positions données dans les registres H (Numéro de colonne) et L (numéro de ligne).

Utilisation du programme

En ce qui concerne le programme Basic, vous n'avez qu'à taper K pour une cassette ou D pour une disquette lors de la demande dy type de support. Lors de la copie de fichier, le programme vous demande le nom du fichier en entrée puis celui du fichier en sortie. Le nom que vous tapez alors doit obéir aux règles suivantes:

Nom du fichier en entrée

Il doit contenir au maximum 16 caractères pour une cassette et 12 (8 pour le nom *et éventuellement 4 pour le type sous la* forme. XXX). Le programme considère le nom, comme fourni lorsque vous avez atteint ce maximum ou appuyé sur ENTER. Ce nom est obligatoirement pour un fichier sur disquette, mais facultatif pour un fichier sur cassette.
Si vous ne fournissez pas de nom, appuyez sur ENTER. Le programme chargera alors le premier fichier trouve sur la cassette. Si il s'agit d'une disquette, vous aurez le message "Bad command" et le programme vous redemandera le nom du fichier.

Nom du fichier en sortie

Il doit obéir aux mêmes règles que le nom du fichier en entrée. Appuyez sur ENTER directement si vous voulez reconfuire le nom du fichier d'entrée sur le fichier de sortie. Toutefois, si vous n'avez pas fourni de nom de fichier d'entrée, le fichier sera écrit sans nom sur la cassette (Unnamed File), et sur disquette vous aurez le message d'erreur "Bad Command" puis à nouveau la demande de nom du fichier en sortie.

Demande d'un autre fichier

Après le traitement complet d'un fichier, le programme vous demandera: AUTRE FICHIER (O/N)?
Si vous avez à traiter un autre fichier obéissant aux mêmes conditions de support, tapez O. Sinon, tapez N. Le programme effectue alors un RESET général (par RST 0).

R.P. Spiegel

```
10 REM ********************************************
20 REM *        Auteur : R.P.SPIEGEL - aout 1985      *
30 REM ********************************************
40 MEMORY 42164:CLEAR:CLS
50 LOCATE 1,1:PRINT "SUPPORT DU FICHIER ENTREE (K ou D) ? "
60 GOSUB 130:LOCATE 38,1:PRINT e$:e1$=e$
70 LOCATE 1,3:PRINT "SUPPORT DU FICHIER SORTIE (K ou D) ? "
80 GOSUB 130:LOCATE 38,3:PRINT e$
90 FOR i=42165 TO 42619:READ a%:POKE i,a%:NEXT
100 IF e1$="K" THEN |TAPE.IN:POKE 42168,0:GOTO 110 ELSE POKE 42168,1
110 IF e$="K" THEN |TAPE.OUT:POKE 42169,1:GOTO 120 ELSE POKE 42169,1
120 CALL 42165
130 e$="":WHILE e$="":e$=INKEY$:WEND:e$=UPPER$(e$):IF e$<>"K" AND e$<>"D" THEN G
OTO 130
140 RETURN
150 DATA &c3,&57,&a5,&00,&00,&4e,&4f,&4d,&20,&44,&55,&20,&46,&49,&43,&48,&49,&45
,&52,&20,&45,&4e,&20,&45,&4e,&54,&52,&45,&45,&20,&3f,&20
160 DATA &4e,&4f,&4d,&20,&44,&55,&20,&46,&49,&43,&48,&49,&45,&52,&20,&45,&4e,&20
,&53,&4f,&52,&54,&49,&45,&20,&3f,&20
170 DATA &41,&55,&54,&52,&45,&20,&46,&49,&43,&48,&49,&45,&52,&20,&28,&4f,&2f,&4e
,&29,&20,&3f,&20
180 DATA &00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00
,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00
190 DATA &00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00
,&00,&00,&00,&00,&00,&00,&00,&00,&00
200 DATA &00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00
,&00,&00
210 DATA &3e,&02,&cd,&0e,&bc,&3e,&00,&06,&00,&0e,&00,&cd,&32,&bc,&3e,&01,&06,&12
,&48,&cd,&32,&bc
220 DATA &cd,54,&bb,&cd,&72,&a6,&21,&ba,&a4,&cd,&48,&a6,&06,&11,&36,&00,&23,&10,
&fb,&21,&06,&a5,&0e,&00,&3a,&b8,&a4
230 DATA &cd,&55,&a6,&21,&16,&a5,&71,&41,&21,&06,&a5,&11,&72,&01,&cd,&77,&bc,&da
,&a2,&a5,&cd,&46,&a6,&c3,&70,&a5
240 DATA &32,&28,&a5,&ed,&53,&29,&a5,&ed,&43,&2d,&a5,&11,&2f,&a5,&01,&18,&00,&09
,&01,&28,&00,&ed,&b0,&2a,&29,&a5,&cd,&83,&bc
250 DATA &22,&2b,&a5,&cd,&7a,&bc,&cd,&72,&a6,&21,&d5,&a4,&cd,&48,&a6,&11,&17,&a5
,&01,&11,&00,&ed,&b0,&21,&17,&a5,&0e,&00
260 DATA &3a,&b9,&a4,&cd,&55,&a6,&79,&fe,&00,&ca,&eb,&a5,&21,&27,&a5,&71,&21,&27
,&a5,&46,&21,&17,&a5,&11,&72,&01,&cd,&8c,&bc
270 DATA &da,&01,&a6,&cd,&46,&a6,&c3,&c5,&a5,&01,&18,&00,&09,&eb,&21,&2f,&a5,&01
,&29,&00,&ed,&b0,&ed,&4b,&2b,&a5,&2a,&29,&a5
280 DATA &ed,&5b,&2d,&a5,&3a,&28,&a5,&cb,&87,&cd,&98,&bc,&cd,&8f,&bc,&cd,&72,&a6
,&21,&f0,&a4,&06,&16,&7e,&cd,&5a,&bb,&23
290 DATA &10,&f9,&cd,&06,&bb,&cd,&5a,&bb,&f6,&20,&fe,&6f,&ca,&70,&a5,&fe,&6e,&c2
,&24,&a6,&c7,&37,&c9,&06,&1b,&7e,&cd,&5a,&bb
300 DATA &23,&10,&f9,&21,&06,&a5,&c9,&fe,&00,&ca,&5f,&a6,&06,&0c,&c3,&61,&a6,&06
,&10,&cd,&06,&bb,&fe,&0d,&ca,&71,&a6,&0c
310 DATA &cd,&5a,&bb,&77,&23,&10,&f0,&c9,&cd,&78,&bb,&26,&01,&2c,&cd,&75,&bb,&c9
```



```
Hisoft GENA3.1 Assembler. Page     1.

Pass 1 errors: 00

A4B5                10          org  #a4b5
A4B5                20          ent  $
A4B5  C357A5        30          jp   debut
A4B8  00            40 typ_in:  defb 0
A4B9  00            50 typ_ou:  defb 0
A4BA  4E4F4D20      60 dem_in:  defm "NOM DU FICHIER EN ENTREE ? "
A4D5  4E4F4D20      70 dem_ou:  defm "NOM DU FICHIER EN SORTIE ? "
A4F0  41555452      80 dem_au:  defm "AUTRE FICHIER (O/N) ? "
A506                90 nom_in:  defs 16              ; nom du fichier en entre
e
A516  00           100 lng_in:  defb 0               ; longueur nom du fichier
 entree
A517               110 nom_ou:  defs 16              ; nom du fichier sortie
A527  00           120 lng_ou:  defb 0               ; longueur nom du fichier
 sortie
A528  00           130 typ_pg:  defb 0               ; type du fichier entree
A529  0000         140 adr_de:  defw 00              ; adresse debut fichier e
n RAM
A52B  0000         150 adr_en:  defw 00              ; adresse point d'entree
A52D  0000         160 lng_pg:  defw 00              ; longueur totale du fich
ier
A52F               170 usr_zo:  defs 40              ; zone utilisateur du lab
el
A557  3E02         180 debut:   ld   a,2
A559  CD0EBC       190          call #bc0e           ; mise de l'ecran en mode
 80 colonnes
A55C  3E00         200          ld   a,0
A55E  0600         210          ld   b,0
A560  0E00         220          ld   c,0
A562  CD32BC       230          call #bc32           ; ink 0 noire
A565  3E01         240          ld   a,1
A567  0612         250          ld   b,#12
A569  48           260          ld   c,b
A56A  CD32BC       270          call #bc32           ; ink 1 verte
A56D  CD54BB       280          call #bb54
A570  CD72A6       290 deminp:  call plcur
A573  21BAA4       300          ld   hl,dem_in
A576  CD48A6       310          call demnom
A579  0611         320          ld   b,#11
A57B  3600         330 razinp:  ld   (hl),#0         ; mise a zero du nom du f
ichier
A57D  23           340          inc  hl
A57E  10FB         350          djnz razinp
A580  2106A5       360          ld   hl,nom_in
A583  0E00         370          ld   c,#0
A585  3AB8A4       380          ld   a,(typ_in)
A588  CD55A6       390          call sainom
A58B  2116A5       400 ouvinp:  ld   hl,lng_in
A58E  71           410          ld   (hl),c
A58F  41           420          ld   b,c
A590  2106A5       430          ld   hl,nom_in
A593  117201       440          ld   de,#172
A596  CD77BC       450          call #bc77           ; OPEN fichier entree
```

```
A599  DAA2A5            jp    c,chrinp
A59C  CD46A6            call  affer1       ; si erreur a l'OPEN
A59F  C370A5            jp    deminp
A5A2  3228A5    chrinp: ld    (typ_pg),a
A5A5  ED5329A5          ld    (adr_de),de
A5A9  ED432DA5          ld    (lng_pg),bc
A5AD  112FA5            ld    de,usr_zo
A5B0  011800            ld    bc,#18
A5B3  09                add   hl,bc
A5B4  012800            ld    bc,#28
A5B7  EDB0              ldir
A5B9  2A29A5            ld    hl,(adr_de)
A5BC  CD83BC            call  #bc83        ; lecture fichier
A5BF  222BA5            ld    (adr_en),hl
A5C2  CD7ABC            call  #bc7a        ; CLOSE fichier entree
A5C5  CD72A6    demout: call  plccur
A5C8  21D5A4            ld    hl,dem_ou
A5CB  CD49A6            call  demnom
A5CE  1117A5            ld    de,nom_ou
A5D1  011100            ld    bc,#11
A5D4  EDB0              ldir
A5D6  2117A5            ld    hl,nom_ou
A5D9  0E00              ld    c,#0
A5DB  3AB9A4            ld    a,(typ_ou)
A5DE  CD55A6            call  sainom
A5E1  79        ouvout: ld    a,c
A5E2  FE00              cp    0
A5E4  CAEBA5            jp    z,memnom
A5E7  2127A5            ld    hl,lng_ou
A5EA  71                ld    (hl),c
A5EB  2127A5    memnom: ld    hl,lng_ou
A5EE  46                ld    b,(hl)
A5EF  2117A5            ld    hl,nom_ou
A5F2  117201            ld    de,#172
A5F5  CD8CBC            call  #bc8c        ; OPEN fichier sortie
A5F8  DA01A6            jp    c,wrtout
A5FB  CD46A6            call  affer1       ; si erreur a l'OPEN
A5FE  C3C5A5            jp    demout
A601  011800    wrtout: ld    bc,#18
A604  09                add   hl,bc
A605  EB                ex    de,hl
A606  212FA5            ld    hl,usr_zo
A609  012800            ld    bc,#28
A60C  EDB0              ldir
A60E  ED4B2BA5          ld    bc,(adr_en)
A612  2A29A5            ld    hl,(adr_de)
A615  ED5B2DA5          ld    de,(lng_pg)
A619  3A28A5            ld    a,(typ_pg)   ; type fichier sortie
A61C  CB87              res   0,a          ; deprotection fichier
A61E  CD98BC            call  #bc98        ; ecriture fichier
A621  CD8FBC            call  #bc8f        ; CLOSE fichier sortie
A624  CD72A6    askaut: call  plccur       ; demande si autre fichie
r
A627  21F0A4            ld    hl,dem_au
A62A  0616              ld    b,#16
A62C  7E        affaut: ld    a,(hl)
A62D  CD5ABB            call  #bb5a
A630  23                inc   hl
```



```
A631  10F9    1030          djnz  affaut
A633  CD06BB  1040          call  #bb06
A636  CD5ABB  1050          call  #bb5a
A639  F620    1060          or    #20
A63B  FE6F    1070          cp    #6f
A63D  CA70A5  1080          jp    z,deminp
A640  FE6E    1090          cp    #6e
A642  C224A6  1100          jp    nz,askaut
A645  C7      1110          ret   0
A646  37      1120 afferl:  scf
A647  C9      1130          ret
A648  061B    1140 demnom:  ld    b,#1b       ; demande nom du fichier
A64A  7E      1150 asknom:  ld    a,(hl)
A64B  CD5ABB  1160          call  #bb5a
A64E  23      1170          inc   hl
A64F  10F9    1180          djnz  asknom
A651  2106A5  1190          ld    hl,nom_in
A654  C9      1200          ret
A655  FE00    1210 sainom:  cp    0           ; saisie nom du fichier
A657  CA5FA6  1220          jp    z,casfic
A65A  060C    1230          ld    b,#c        ; longueur maxi 12 si fic
hier diskette
A65C  C361A6  1240          jp    lirnom
A65F  0610    1250 casfic:  ld    b,#10       ; longueur maxi 16 si fic
hier cassette
A661  CD06BB  1260 lirnom:  call  #bb06
A664  FE0D    1270          cp    13
A666  CA71A6  1280          jp    z,retsai
A669  0C      1290          inc   c
A66A  CD5ABB  1300          call  #bb5a
A66D  77      1310          ld    (hl),a
A66E  23      1320          inc   hl
A66F  10F0    1330          djnz  lirnom
A671  C9      1340 retsai:  ret
A672  CD78BB  1350 plccur:  call  #bb78       ; remise du curseur col 1
 ligne suivante
A675  2601    1360          ld    h,1
A677  2C      1370          inc   l
A678  CD75BB  1380          call  #bb75
A67B  C9      1390          ret

Pass 2 errors: 00

adr_de A529  adr_en A52B  affaut A62C
afferl A646  askaut A624  asknom A64A
casfic A65F  chrinp A5A2  debut  A557
dem_au A4F0  dem_in A4BA  dem_ou A4D5
deminp A570  demnom A648  demout A5C5
lirnom A661  lng_in A516  lng_ou A527
lng_pg A52D  memnom A5EB  nom_in A506
nom_ou A517  ouvinp A59B  ouvout A5E1
plccur A672  razinp A57B  retsai A671
sainom A655  typ_in A4B9  typ_ou A4B9
typ_pg A528  usr_zo A52F  wrtout A601

Table used:   441  from   500
Executes: 42165
```

Initiation à l'ASSEMBLEUR

Vous êtes fatigué d'attendre devant votre écran le résultat d'un tri ; la torpille nucléaire de votre dernière réalisation de jeu d'arcade se déplace à la vitesse foudroyante d'un escargot rhumatisant ; vous êtes excédé par la lenteur du Basic et l'absence de fonctions performantes...
Alors, pas de doute, vous êtes candidat à l'étude de l'assembleur. Il est évident que pour aborder cette matière, de bonnes notions de Basic et d'architecture interne de l'ordinateur sont nécessaires. Les notions de variable, de boucle, de saut ou de tableau (DIM) ne doivent pas être étrangères au candidat à l'étude de l'assembleur. Les prodiges réalisés par la programmation en assembleur valent largement l'effort consenti pour apprendre ce langage.

L'assembleur est considérablement plus complexe que le Basic. Par contre, il vous permet le contrôle total de votre système avec une flexibilité inégalée dans la manipulation des données, une vitesse d'exécution accrue, une réduction de la taille de certains programmes et enfin, la satisfaction de pouvoir dire à vos amis : non, ce n'est pas du Basic, je l'ai écrit moi-même en assembleur.

Généralités

La démarche permettant d'écrire un programme en Assembleur est fondamentalement différente de celle entreprise en vue de réaliser un programme Basic. Dans votre micro-ordinateur, le langage Basic est dit : INTERPRÈTE. Cela signifie que les lignes de programme sont lues, analysées, transformées en code machine et exécutées les unes après les autres. Elles sont successivement interprêtées par la machine grâce à un programme interne appelé « interpréteur Basic ». pour travailler en Basic, il suffit donc d'écrire un programme dans la mémoire centrale et de donner l'ordre « RUN » à l'ordinateur.
Pour l'assembleur, les choses ne sont pas aussi simples. En effet, l'assembleur n'est pas un langage interprété mais bien un langage compilé. Avant de lancer l'exécution d'un programme assembleur, il faut le compiler. Cette opération consiste à traduire l'entier du programme en code machine. Le résultat de la traduction en code machine devient un programme exécutable autant de fois que l'utilisateur le désire et appelé programme objet. Le programme écrit en langage assembleur et non encore compilé s'appelle programme source.
Le programme objet est uniquement composé d'une suite de codes machine et il est pratiquement impossible à déchiffrer. Par contre, le programme source peut être relu et modifié sans problèmes. Il suffira de recompiler ce dernier après modifications afin de créer un nouveau programme objet exécutable.
Pour écrire un programme source, on utilise un éditeur. Ensuite, il suffit de compiler le programme source à l'aide de l'assembleur afin d'obtenir le programme objet. Ces deux programmes (l'éditeur et l'assembleur) font souvent partie d'un même logiciel appelé éditeur-assembleur.
Il ne faut donc pas confondre le langage assembleur avec le programme permettant la compilation du programme source, également appelé assembleur.

programme source → assembleur → programme objet

Introduction

Pour commencer, examinons la démarche à suivre à l'aide d'un exemple simple : soit l'instruction Basic Z = B + C. Pour l'écrire en assembleur, il sera nécessaire de :
1. Pointer la valeur de B et la placer dans l'accumulateur. L'accumulateur constitue l'endroit privilégié de traitement de votre microprocesseur. On peut le considérer comme un « tiroir » dans lequel on dépose les objets à traiter.
2. Positionner un pointeur sur la variable C.
3. Additionner au contenu de l'accumulateur le contenu de l'endroit pointé (C).
4. Sauver le nouveau contenu de l'accumulateur à l'adresse devant contenir la variable Z.
La lecture de ces quelques lignes montre à quel point le langage assembleur est proche de la fonction primaire de l'ordinateur. En effet, les différentes actions permises par ce langage sont élémentaires. C'est en ayant la possibilité de réaliser ses applications au niveau élémentaire que le programmeur pourra obtenir de sa machine les performances désirées et impossibles en Basic. Le petit exemple ci-dessus souffre d'ailleurs de grosses restrictions. En effet, si la précision de l'opération Basic Z = B + C ne dépend que du type de variable (entière ou flottante), en assembleur, le programme décrit ne fonctionne que pour des variables Z, B et C dont les valeurs sont comprises entre 0 et 255.
Chaque instruction décrite en langage assembleur est traduite en langage machine par un utilitaire spécialisé appelé ASSEMBLEUR. En langage machine, chaque instruction est composée d'une suite de 0 et de 1. Par exemple : 00111110. Si cette suite prend tout son sens pour un ordinateur, il faut avouer qu'elle n'est pas évidente pour le cerveau humain qui préfère globaliser les concepts sous forme de mots. C'est la raison d'être du langage assembleur. En effet, il permet au programmeur de remplacer les différentes instructions élémentaires par des abréviations. Ces abréviations portent le nom de mnémoniques.
Hélas, les premiers concepteurs de ce langage étant anglo-saxons, c'est dans cette langue que les mnémoniques ont été écrites. Par exemple, CHARGER se dit LOAD en anglais ; le mnémonique correspondant sera donc LD.
Pour terminer les généralités, rappelons que chaque ordinateur est équipé d'un cœur, c'est-à-dire d'un organe de calcul et de traitement. Ce dernier est constitué d'un circuit appelé microprocesseur.
Le microprocesseur est un composant électronique muni de 40 « pattes ». Il se présente sous la forme d'un petit parallélépipède rectangle très plat de 6 centimètres sur 2.
De nombreux types de microprocesseurs existent présentant chacun divers avantages et inconvénients. Le langage Assembleur étant très proche du microprocesseur, il sera différent suivant le type de microprocesseur utilisé. Celui que nous vous proposons d'étudier s'appelle Z80. Il équipe des ordinateurs aussi divers que l'AMSTRAD, les MSX, les TRS-80 de chez TANDY... C'est l'un des processeurs les plus célèbres et malgré son grand âge (8 ans), il est considéré comme l'un des meilleurs processeurs de sa catégorie.

Le Z80 est un microprocesseur dit de 8 bits. Cela signifie qu'il est capable de traiter 8 états 0 ou 1 simultanément. Une confusion règne dans les esprits même les plus éclairés depuis l'apparition du microprocesseur 16 bits. Pour simplifier les choses, il faut savoir qu'il existe trois critères permettant de classifier les différents types de microprocesseur :
1) Le nombre de bits de données que le microprocesseur est capable de lire simultanément ;
2) Le nombre de bits que le microprocesseur est capable de traiter simultanément ;
3) Le nombre de bits permettant d'adresser un endroit quelconque situé dans l'espace mémoire.
Le Z80 est un microprocesseur capable de lire et de traiter simultanément 8 bits de données (1 et 2). Il est équipé d'un bus d'adresse de 16 bits permettant d'adresser 65536 emplacements mémoire différents (16 états 0 ou 1 permettant 65536 possibilités).

Et les variables ?

En assembleur, il n'existe pas d'identificateur de variable. En Basic, ou dans tout autre langage évolué, chaque variable porte un nom qui permet de retrouver son contenu. Le langage évolué s'occupe de placer le contenu de la variable à un emplacement mémoire et d'aller le récupérer lorsque cela est nécessaire.
Considérons le programme Basic suivant :

```
10 A=4
20 B=5
30 C=A+B
40 PRINT C
```

En Basic, les différents emplacements mémoire où sont stockés les contenus des variables A, B et C ainsi que l'endroit où s'effectue l'addition n'ont aucune importance pour le programmeur. En assembleur, ils sont au contraire très importants. En effet, l'assembleur n'utilise pas de variables. Les différentes valeurs utilisées par le programme doivent être écrites directement aux adresses mémoire adéquates. Autrement dit, tout doit être parfaitement localisé puisqu'on travaille directement dans la mémoire.
Quant au microprocesseur, il possède sa propre petite mémoire interne. Les différents emplacements de sa mémoire interne portent le nom de registres et sont identifiées chacun par une lettre. Ils servent à contenir les données sur lesquelles le programmeur désire effectuer différentes opérations et à effectuer ces dernières.
Le diagramme suivant représente les différents registres du Z80 :

8 bits	8 bits
A	F
B	C
D	E
H	L

8 bits	8 bits
A'	F'
B'	C'
D'	E'
H'	L'

16 bits	
IX	
IY	
I	R

16 bits
PC
SP

Le registre A constitue la réponse à la question « Où l'addition a-t-elle lieu ? » En effet, le registre A, appelé accumulateur, contient toujours une des deux opérandes dans les instructions d'addition, de soustraction, de fonction logique... Le résultat de l'opération effectuée se retrouve toujours dans ce registre.
Le registre F, ou FLAG (FLAG signifie drapeau en anglais), est un registre particulier où sont mémorisés les indicateurs de dépassement de capacité, de valeur 0, de valeur négative... Les autres registres sont généralement utilisés comme pointeurs vers les différentes adresses de la mémoire utilisées.
Remarque : les registres B, C, D, E, H et L sont des registres 8 bits. Or le bus d'adresse du Z80 est constitué de 16 bits. Un seul de ces registres ne peut donc suffire à adresser distinctement un emplacement mémoire. Ils sont donc

couplés pour former les trois registres de 16 bits suivants : BC, DE et HL. Les registres PC et SP sont deux registres de 16 bits. Le PC, ou Program Counter (compteur de programme), pointe toujours vers une adresse mémoire. Cette adresse contient une indication permettant de connaître l'endroit précis où se déroule le programme à l'instant t. Le PC contient donc une adresse qui fonctionne de façon similaire au pointeur de la ligne Basic en cours d'exécution. Le SP, ou Stack Pointer (pointeur de pile), est un registre très important. Il contient l'adresse de la pile (stack). Le programmeur peut positionner la pile là où il le désire mais il a tout intérêt à la disposer en haut de la mémoire.

A présent, voyons à quoi sert cette pile et ce qu'elle peut bien contenir :

Une pile (du verbe empiler, comparez à une pile d'assiettes) constitue une collection d'objets où des ajouts et des retraits peuvent être effectués. Il existe deux sortes de piles : la pile LIFO et la pile FIFO :

LIFO : de l'anglais Last in First out, signifie « dernier entré, premier sorti ». Cette pile peut se comparer à une pile d'assiettes. La dernière assiette ajoutée à la pile sera la première que l'on prendra lorsqu'un retrait sera nécessaire.

FIFO : de l'anglais First in First out, signifie « premier entré, premier sorti ». La meilleure comparaison pour cette pile est l'appareil qui tombe le premier dans les mains de l'acheteur.

La pile de Z80 est une pile LIFO. Seul son sommet est accessible (pour prendre la troisième assiette, il faut d'abord enlever les deux premières). Les autres registres sont plus particuliers et ne seront pas abordés au cours de ce premier article.

La synthaxe et la notation

En Basic, lorsque nous décrivons une variable, nous parlons de son contenu. L'endroit où elle est stockée n'a aucune importance puisqu'elle a un nom. Lorsqu'on donne le nom d'une variable Basic, c'est donc bien pour utiliser son contenu.

En assembleur, les variables n'ont pas de nom. Elles sont connues uniquement par leur adresse. Il faut donc faire la distinction entre une adresse et la valeur qui s'y trouve stockée. Si ADR représente une adresse, nous indiquerons ADR pour parler de l'adresse et (ADR) pour parler de son contenu. Rappelons ici que ADR est un nombre de 16 bits et que (ADR) est un nombre de 8 bits. Pour le microprocesseur, une instruction est une suite de bits appelée code opératoire. Les valeurs des suites de bits représentant les différents codes opératoires sont généralement représentées en notation hexadécimale. Comme le cerveau humain n'aime pas beaucoup manipuler des chiffres, les codes opératoires seront représentés par des mnémoniques correspondant aux notations hexadécimales.

Pour chacune des instructions que nous allons décrire, nous donnerons le code opératoire en hexadécimal, la mnémonique (ou opérateur) correspondante et le format des ... Par convention, A, B, C, D, E, H, L et SP identifient les registres correspondants du microprocesseur ; adr signifie adresse ; (adr) identifie le contenu de l'adresse adr ; r signifie registre ; n signifie constante sur 8 bits ; (r) indique le contenu de l'adresse contenue dans le registre r.

Les instructions de base

Le microprocesseur peut recopier le contenu d'une de ses mémoires internes (registre) dans une autre. Cette opération s'appelle une copie de registre à registre.

Le format général de cette opération est le suivant :

LD r, r'

où LD est la mnémonique ou opérateur et r et r', les opérandes. r et r' peuvent être n'importe quel registre parmi A, B, C, D, E, H et L.

L'instruction LD (load en anglais signifie charger) copie le contenu du registre r' dans le registre r.

Chaque combinaison de l'instruction Ld avec deux registres possède son propre code opératoire. Le code opératoire est constitué d'un simple octet (8 bits) représenté par une valeur hexadécimale à deux chiffres.

Le tableau suivant donne la liste des différents codes opératoires pour l'instruction LD :

En observant ce tableau, nous constatons que les codes opératoires suivent une logique évidente. Le trou entre L et A sera comblé par la suite. Il est évident que les instructions de type LD A,A ; LD H,H..., correspondant à la diagonale 40 à 7F, ne présentent aucun intérêt.

Par exemple : copier le contenu du registre H dans l'accumulateur A s'écrira :

LD A,H

L'assembleur fournira le code opératoire 7C.

Autre exemple : copier le contenu de la paire de registre BC dans la paire de registres DE s'écrira :

LD D,B
LD E,C

L'assembleur fournira la suite de codes opératoires 50,59.

Il existe une méthode plus courante permettant de copier les registres. Elle utilise la pile. Il suffit de « pousser » le contenu du registre BC dans la pile et d'ensuite retirer le contenu de la pile dans DE. Cette technique sera expliquée par la suite.

Les différents emplacements mémoire contiennent des valeurs d'une taille de 8 bits. Ces valeurs peuvent être chargées de la mémoire vers un registre et vice-versa.

Les mouvements entre la mémoire et les registres sont très limités. Le registre HL est le plus utilisé pour ce type d'opération. En effet, le jeu d'instruction de HL est plus riche que ceux de BC et de DE. Bien entendu, par l'intermédiaire de HL, on peut charger le contenu d'une mémoire dans n'importe quel registre et vice versa. Les instructions utilisées à cet effet sont les suivantes :

LD r, (HL)
et
LD (HL), r

où r représente n'importe quel registre 8 bits (A, B, C, D, E, H et L).

Les codes opératoires correspondants sont fournis par le tableau suivant :

	B	C	D	E	H	L		A
B	40	41	42	43	44	45		47
C	48	49	4A	4B	4C	4D		4F
D	50	51	52	53	54	55		57
E	58	59	5A	5B	5C	5D		5F
H	60	61	62	63	64	65		67
L	68	69	6A	6B	6C	6D		6F
A	78	79	7A	7B	7C	7D		7F

	B	C	D	E	H	L	A
r, (HL)	46	4E	56	5E	66	6E	7E
(HL), r	70	71	72	73	74	75	77

Vous pouvez, à l'aide de ce tableau, compléter le trou du précédent. Seule la case de croisement correspondant à :

LD (HL), (HL)

n'a pas été décrite. Son code opératoire est 76. Au lieu de recopier le contenu de l'adresse pointée par HL sur lui-même, le code opératoire 76 arrête le microprocesseur (HALT).
D'autres méthodes sont permises pour effectuer des mouvements entre l'accumulateur et la mémoire :

1°) Mouvement entre A et un emplacement mémoire dont l'adresse est contenue dans une des paires de registres BC ou DE.

Syntaxe	code opératoire
LD A, (BC)	0A
LD A, (DE)	1A
LD (BC), A	02
LD (DE), A	12

2°) Mouvement entre A et un emplacement mémoire indiqué par une adresse en clair :

Syntaxe	code opératoire
LD A, (adr)	3A VW XY
LD (adr), A	32 VW XY

où VW et XY sont les valeurs hexadécimales représentant l'adresse réelle de la mémoire avec la convention suivante :
VW représente l'octet de poids faible de l'adresse mémoire et XY représente l'octet de poids fort.

Exemple : LD A, (3B4C) charge dans l'accumulateur le contenu de la mémoire 3B4C et produit comme code opération 3A 4C 3B.
Cette inversion entre le poids fort et le poids faible de l'adresse à l'intérieur du code opératoire se retrouvera dans toutes les instructions portant sur une adresse directe mémoire.

Les chargements immédiats

Il est évident qu'il est indispensable de pouvoir charger une constante directement dans l'accumulateur ou dans un registre 8 bits ainsi que de pouvoir charger une adresse directement dans une paire de registres (BC, DE et HL).

Syntaxe	code opératoire
LD r,n	voir tableau
LD (HL), n	36 n
LD HL, adr	21 VW XY
LD DE, adr	11 VW XY
LD BC, adr	01 VW XY

	B	C	D	E	H	L	A
LD r, n	06 n	0E n	16 n	1E n	26 n	2E n	3E n

REMARQUE : pour toutes les instructions vues jusqu'à présent, une seule mnémonique a été utilisée : LD.
D'autres possibilités existent pour charger des registres de 16 bits ou pour la technique du chargement indexé. Elles dépassent le cadre de cet article. En voici la syntaxe et les codes opératoires :

syntaxe des instructions indexées :
1 - LD r, (IX+n)
2 - LD r, (IY+n)
3 - LD (IX+n), r
4 - LD (IY+n), r

code opératoire :

	B	C	D	E	H	L	A
1—	DD 46 n	DD 4E n	DD 56 n	DD 5E n	DD 66 n	DD 6E n	DD 7E n
2—	FD 46 n	FD 4E n	FD 56 n	FD 5E n	FD 66 n	FD 6E n	FD 7E n
3—	DD 70 n	DD 71 n	DD 72 n	DD 73 n	DD 74 n	DD 75 n	DD 77n
4—	FD 70 n	FD 71 n	FD 72 n	FD 73 n	FD 74 n	FD 75 n	FD 77 n

Instructions 16 bits	codes opératoires
LD SP, HL	F9
LD SP, IY	DD F9
LD SP, IX	FD F9
LD HL, (adr)	2A VW XY
LD (adr), HL	22 VW XY
LD BC, (adr)	ED 4B VW XY
LD (adr), BC	ED 43 VW XY
LD DE, (adr)	ED 5B VW XY
LD (adr), DE	ED 53 VW XY
LD SP, (adr)	ED 7B VW XY
LD (adr), SP	ED 73 VW XY
LD IX, (adr)	DD 2A VW XY
LD (adr), IX	DD 22 VW XY
LD IY, (adr)	FD 2A VW XY
LD (adr), IY	FD 22 VW XY

Les instructions de manipulation de pile

Pour manipuler la pile, deux instructions sont à notre disposition :
PUSH : permet de pousser une paire de registres dans le pile,
POP : permet de pousser le sommet de la pile dans une paire de registres.
A la suite d'unPUSH ou d'un POP, l'adresse de la pile contenue dans le registre SP est respectivement décrémentée ou incrémentée de deux octets. En effet, la pile se trouvant en haut de mémoire, chaque fois qu'on y ajoute quelque chose (PUSH), son adresse de début diminue.

	AF	BC	DE	HL	IX	IY
PUSH	F5	C5	D5	E5	DD E5	FD E5
POP	F1	C1	D1	E1	DD E1	FD E1

Comme nous l'avons laissé entrevoir précédemment, pour copier le contenu de BC dans DE, on peut remplacer :

LD D, B
LD E, C

par :

PUSH BC
POP DE

Pour terminer la liste des instructions portant sur les adresses et le chargement, voyons l'instruction permettant de permuter les contenus de certains registres : EX.

syntaxe	code opératoire
EX DE, HL	EB
EX (SP), HL	E3
EX (SP), IX	DD E3
EX (SP), IY	FD E3

EXERCICES

1) Ecrivez le programme assembleur qui échange les contenus de B et de C. Donnez les codes opératoires correspondants.

2) Si le contenu de la mémoire est le suivant :

adresse	contenu
8036	00
8037	08
8038	10
8039	18
803A	20
803B	28
803C	30
803D	38
803E	40
803F	48

a) que contiendra l'accumulateur après la séquence suivante :

```
LD HL, 803D
LD L, (HL)
LD A, (HL)
```

b) écrivez le programme objet correspondant (codes opératoires).

La solution des exercices se trouve en fin d'article.

Instructions arithmétiques et logiques

Après avoir examiné les différentes instructions de manipulation de données, voyons à présent les principales instructions arithmétiques et logiques.
Ces opérations portent toujours sur l'accumulateur et un autre registre ou une constante.
L'opérande sera symbolisée par op. Elle peut prendre les formes suivantes :
— Un des registres A, B, C, D, E, H ou L.
— Une constante n.
— Le contenu de la mémoire pointée par HL ⟶ (HL).
— Un adressage indexé par rapport à IX/IY ⟶ (IX + n)/(IY + n).
Il est très important d'analyser l'effet de ces instructions sur le registre F. Le registre F contient 5 bits importants. Ils sont identifiés par une lettre :
— C = bit de report (Carry).
— Z = bit de Zéro.
— V = bit de dépassement (oVerflow).
— S = bit de Signe.
— P = bit de Parité (nombre de bits à 1 pair ou impair).

Addition sans report

Syntaxe : ADD A, op.
Codes opératoires :

B	C	D	E	H	L	(HL)	A	n	(IX+n)	(IY+n)
80	81	82	83	84	85	86	87	C6n	DD86n	FD86n

Après l'exécution de l'instruction, A contient la somme de l'ancienne valeur de A et du contenu de l'opérande spécifiée.

Addition avec report

Syntaxe : ADC A, op.
Codes opératoires :

B	C	D	E	H	L	(HL)	A	n	(IX+n)	(IY+n)
88	89	8A	8B	8C	8D	8E	8F	CEn	DD8En	FD8En

Après cette instruction, le contenu de A vaut la somme de l'ancien contenu de A, du contenu de l'opérande spécifiée et du bit de report de l'opération précédente s'il y en a un.
Dans les deux instructions précédentes, les indicateurs C, Z, V et S sont positionnés en fonctions du résultat de l'opération.

Soustraction avec et sans report

La soustraction fonctionne selon le même principe que l'addition.
Syntaxe : SUB op (soustraction sans report)
SBC A, op (avec report).
Codes opératoires :

	B	C	D	E	H	L	(HL)	A	n	(IX+n)	(IY+n)
SUB	90	91	92	93	94	95	96	97	D6n	DD96n	FD96n
SBC	98	99	9A	9B	9C	9D	9E	9F	DEn	DD9En	FD9En

Remarque : La multiplication et la division ne font pas partie des instructions de base du Z80, elles doivent être simulées par une série d'additions ou de soustractions ou de manipulations de bits.

Les opérations logiques AND, OR et XOR

L'action de ces instructions qui portent sur l'accumulateur et une opérande identique à celles décrites pour l'addition est similaire à celles des fonctions BASIC correspondantes. Le résultat de l'opération se retrouve dans l'accumulateur.

Table de vérité :

A	opérande	AND	OR	XOR
0	0	0	0	0
0	1	0	0	0
1	0	0	1	1
1	1	1	1	0

Les indicateurs C, Z, P et S sont affectés par ces opérations. L'indicateur C est systématiquement remis à zéro.

Syntaxe : AND op
OR op
XOR op

Codes opératoires :

	B	C	D	E	H	L	(HL)	A	n	(IX+n)	(IY+n)
AND	A0	A1	A2	A3	A4	A5	A6	A7	E6n	DDA6n	FDA6n
XOR	A8	A9	AA	AB	AC	AD	AE	AF	EEn	DDAEn	FDAEn
OR	B0	B1	B2	B3	B4	B5	B6	B7	F6n	DDB6n	FDB6n

L'instruction de comparaison

L'opération de comparaison est une simple soustraction entre l'accumulateur et l'opérande spécifiée. Les indicateurs sont positionnés en fonction du résultat mais l'accumulateur n'est pas modifié.
Si les 8 bits sont en format non signé, on a la table suivante :

relation	Z	C
A < op	0	1
A = op	1	0
A > op	0	0

Syntaxe : CP op

Codes opératoires :

B	C	D	E	H	L	(HL)	A	n	(IX+n)	(IY+n)
B8	B9	BA	BB	BC	BD	BE	BF	FEn	DD BE n	FD BE n

Il existe d'autres opérations arithmétiques et logiques. En voici une brève description :

— INCREMENTATION (ajouter 1) : INC
— DECREMENTATION (soustraire 1) : DEC

	B	C	D	E	H	L	(HL)	A	IX	IY
INC	04	0C	14	1C	24	2C	34	3C	DD 34 n	FD 34 n
DEC	05	0D	15	1D	25	2D	35	3D	DD 35 n	FD 35 n

— Complémentation de l'ACCUM (inversion des bits :
CPL ED 44

— Complémentation et mise à 1 de l'indicateur de report (CARRY) :
CCF (clear carry flag 3F
SCF (set carry flag) 37

— Instruction sans effet : NOP 00

— On trouve également des instructions d'addition, d'incrémentation et décrémentation entre les registres 16 bits :

	BC	DE	HL	SP	IX	IY
ADD HL,	09	19	29	39	—	—
INC	03	13	23	33	DD 23	FD 23
DEC	0B	1B	2B	3B	DD 2B	FD 2B

EXERCICE

Si la mémoire contient :

adresse	contenu
8000	12
8001	24
8002	28

Ecrire un programme qui sauve à l'adresse 8003 la somme des contenus des deux premières mémoires (8000 et 8001) moins le contenu de la troisième. Utilisez HL pour pointer sur la mémoire.
Ensuite, écrivez le code objet produit par le programme.

Sauts et sous-routines

L'assembleur ne possède pas d'instruction équivalente au IF...THEN...ELSE et au FOR...NEXT du Basic. Il est par contre doté d'instructions équivalentes aux GOTO et GOSUB.
Un IF...THEN sera simulé en assembleur par une instruction de comparaison (CP) suivie d'une instruction de saut.

Une instruction FOR...NEXT sera simulée par une instruction d'incrémentation ou décrémentation suivie d'un test (CP) et d'un saut.

Les sauts

Il existe quatre types de saut. On distingue des sauts ABSOLUS et RELATIFS ; CONDITIONNELS et INCONDITIONNELS.

1. Saut absolu conditionnel
Syntaxe : JP adr
Code opératoire : C3 VW XY
A la suite de l'instruction JP adr, le PC se retrouve à l'adresse adr. Le programme continuera donc à cette adresse.
2. Saut absolu conditionnel.

		syntaxe	code opératoire
saut si non zéro	:	JP NZ, adr	C2 VW XY
Saut si zéro	:	JP Z, adr	CA VW XY
saut si report	:	JP C, adr	DA VW XY
saut si non report	:	JP NC, adr	D2 VW XY
saut si négatif	:	JP M, adr	FA VW XY
saut si positif	:	JP P, adr	F2 VW XY
saut si parité paire	:	JP PE, adr	EA VW XY
saut si parité impaire	:	JP PO, adr	E2 VW XY

3-4. Sauts relatifs conditionnels et inconditionnels.
Au lieu d'indiquer une adresse de saut absolue, on peut indiquer un déplacement en nombre d'octets par rapport à la position courante du PC. Ce type d'opération est appelée saut relatif.
Les déplacements peuvent être compris entre 0 et 127 octets plus loin (00-7F) ou 1 et 128 octets avant la position courante du PC (FF à 80). Cela signifie que le huitième bit de l'octet indiquant le déplacement est un bit de signe.
Cette opération est l'une des plus fastidieuses à calculer à la main. C'est dans le calcul des déplacements que l'utilité d'un ASSEMBLEUR se fait sentir.
L'utilisateur d'un assembleur indique simplement l'endroit où il désire sauter par une étiquette (LABEL). Il fait alors figurer cette étiquette à la place de l'opérande. Ainsi un programme qui additionne les 10 premiers nombres d'une table pointée par HL s'écrira de la façon suivante :

```
        LD      HL, TABLE
        LD      B, 10
SUITE : ADD     A, (HL)
        INC     HL
        DEC     B
        JR      NZ, SUITE
        .
        .
        .
```

L'assemblage manuel se calculera comme suit :

```
        LD      HL, TABLE     21 XX XX
        LD      B, 10         06, 0A
SUITE : ADD     A, (HL)       86
        INC     HL            23
        DEC     B             05
        JR      NZ, SUITE     20 YY
```

Le problème consiste à déterminer la valeur de YY. Après avoir rencontré l'octet 20 suivi de l'octet YY, le PC se trouve déjà à l'adresse suivante. Pour revenir sur YY, il faut faire —1 et afin de pointer sur le 20, il faut faire —2 et

ainsi de suite jusqu'à remonter à 86. Au total cela fait —5. En binaire signé, —5 vaut FB.
JR NZ, SUITE sera traduit par 20 FB
syntaxes des différents sauts relatifs avec leur code opératoire :

JR déplacement	18 XX
JR Z, déplacement	28 XX
JR NZ, déplacement	20 XX
JR C, déplacement	38 XX
JR NC, déplacement	30 XX

Le grand avantage des sauts relatifs par rapport aux sauts absolus est que les programmes qui les utilisent sont indépendants de leur adresse d'implantation dans la mémoire.

Les sous-routines

La notion de sous-routines assembleur est très proche de celle utilisée en Basic. L'appel d'une sous-routine et son retour peuvent être conditionnel ou inconditionnel.

syntaxe		code opératoire
APPEL		
CALL	adr	CD VW XY
CALL	Z, adr	CC VW XY
CALL	NZ, adr	C4 VW XY
CALL	C, adr	DC VW XY
CALL	NC, adr	D4 VW XY
CALL	M, adr	FC VW XY
CALL	P, adr	F4 VW XY
CALL	PE, adr	EC VW XY
CALL	PO, adr	E4 VW XY
RETOUR		
RET		C9
RET	Z	C8
RET	NZ	C0
RET	C	D8
RET	NC	D0
RET	M	F8
RET	P	F0
RET	PE	E8
RET	PO	E0

Lors de l'appel d'une sous-routine, l'adresse de retour est poussée dans la PILE, la plus grande prudence s'impose donc lors de la manipulation de la pile (PUSH POP).

EXERCICE

Si l'adresse ADR contient une constante N et l'adresse ADR+1 contient une constante M, faite un programme qui appelle la sous-routine POSIT si N>M et NEGAT si N<M ou qui saute à l'adresse ADR+2 si N=M.

En guise de conclusion provisoire

Les différentes notions vue jusqu'ici doivent vous permettre si pas d'écrire, au moins de comprendre les programmes simples écrits en assembleur.

Au cours de ces quelques pages, nous n'avons fait que découvrir le sommet de l'iceberg. Il reste encore un bon nombre d'instructions à étudier.
Pour bien programmer en assembleur, une bonne connaissance de l'EDITEUR/ASSEMBLEUR utilisé est indispensable.

Pour celui qui n'en possède pas encore et désire en faire l'acquisition prochainement, nous vous conseillons vivement l'achat du DEVPAC. En effet, il présente des caractéristiques exceptionnelles tout en étant d'une grande simplicité à utiliser.
Enfin, pour bien maîtriser l'assembleur, une connaissance parfaite du logiciel interne de votre ordinateur est indispensable. A ce propos, les ouvrages de la série « CLEFS POUR... » (PSI éditions) semblent particulièrement indiqués. les ouvrages intitulés « LE LIVRE DU... » (BCM éditions, PSI diffusion) vous fourniront tous les renseignements concernant l'architecture interne de votre machine accompagnés de nombreux exemples de programmes assembleur abondamment commentés.

SOLUTIONS DES EXERCICES

```
1) 79  LD A,C
   48  LD C,B
   47  LD B,A
```

```
2) 21 3D 80 LD HL, 803 D
   6E       LD L, (HL)
   7E       LD A, (HL)
```

Au départ, 803 D contient 38.
Après LD L, (HL), L contient aussi 38.
Le contenu de HL devient donc 8038.
LD A, (HL) a pour effet de mettre dans A le contenu de l'adresse pointée par HL, c'est-à-dire le contenu de l'adresse 8038 qui est 10.

```
3) 21 00 80  LD HL, 8000
   86        ADD A, (HL)
   23        INC HL
   86        ADD A, (HL)
   23        INC HL
   96        SUB (HL)
   23        INC HL
   77        LD (HL), A
```

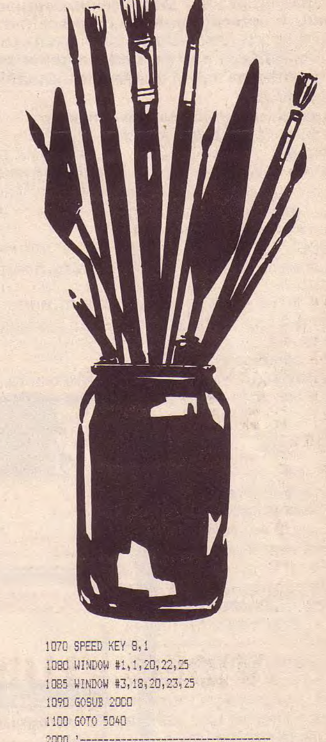
```
4) LD HL, ADR
   LD A, (HL)
   INC HL
   CP (HL)
   JP Z, ADR+2
   CALL NC, POSIT
   CALL NEGAT
```

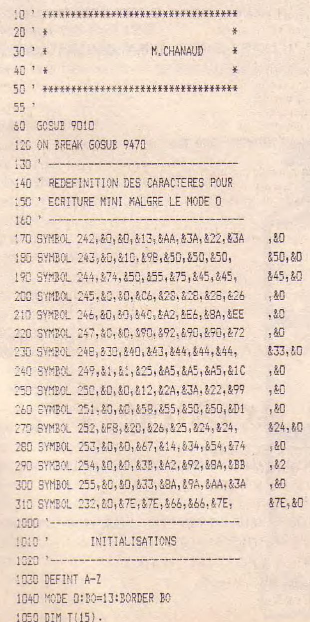


CPC PAINT

CPC PAINT est un utilitaire de création graphique qui permet de se servir au mieux des possibilités graphiques et couleurs de votre CPC. A noter : le chargement d'un écran est possible en dehors de la présence du programme (pour une page de présentation par exemple) il faut taper :
MODE O: WINDOW O, 1,20,25:LOAD" ", COOO (ENTER)

Michel Chanaud





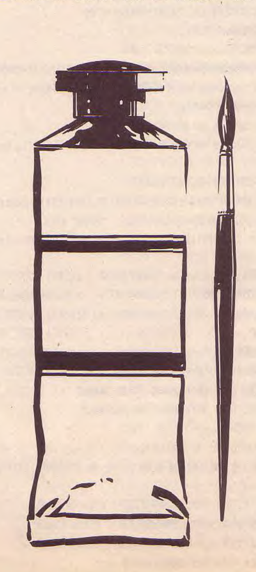
```
10 ' *********************************
20 ' *                               *
30 ' *                M.CHANAUD      *
40 ' *                               *
50 ' *********************************
55 '
60  GOSUB 9010
120 ON BREAK GOSUB 9470
130 ' --------------------------------
140 ' REDEFINITION DES CARACTERES POUR
150 ' ECRITURE MINI MALGRE LE MODE O
160 ' --------------------------------
170 SYMBOL 242,&0,&0,&13,&AA,&3A,&22,&3A    ,&0
180 SYMBOL 243,&0,&10.&98,&50,&50,&50,      &50,&0
190 SYMBOL 244,&74,&50,&55,&75,&45,&45,     &45,&0
200 SYMBOL 245,&0,&0,&C6,&28,&28,&28,&26    ,&0
210 SYMBOL 246,&0,&0,&4C,&A2,&E6,&8A,&EE    ,&0
220 SYMBOL 247,&0,&0,&90,&92,&90,&90,&72    ,&0
230 SYMBOL 248,&30,&40,&43,&44,&44,&44,     &33,&0
240 SYMBOL 249,&1,&1,&25,&A5,&A5,&A5,&1C    ,&0
250 SYMBOL 250,&0,&0,&12,&2A,&3A,&22,&99    ,&0
260 SYMBOL 251,&0,&0,&58,&55,&50,&50,&D1    ,&0
270 SYMBOL 252,&F8,&20,&26,&25,&24,&24,     &24,&0
280 SYMBOL 253,&0,&0,&67,&14,&34,&54,&74    ,&0
290 SYMBOL 254,&0,&0,&3B,&A2,&92,&8A,&BB    ,&2
300 SYMBOL 255,&0,&0,&33,&8A,&9A,&AA,&3A    ,&0
310 SYMBOL 232,&0,&7E,&7E,&66,&66,&7E,      &7E,&0
1000 '--------------------------------
1010 '        INITIALISATIONS
1020 '--------------------------------
1030 DEFINT A-Z
1040 MODE 0:BO=13:BORDER BO
1050 DIM T(15).
1060 P=3:C=0:L=2:CE=0:N=16:DR=0
```

```
1070 SPEED KEY 8,1
1080 WINDOW #1,1,20,22,25
1085 WINDOW #3,18,20,23,25
1090 GOSUB 2000
1100 GOTO 5040
2000 '--------------------------------
2010 ' AFFICHAGE DE LA CHARTE COULEURS
2020 '--------------------------------
2030 CLG
2040 FOR I=3 TO 15:INK I,I-1:T(I)=I-1:NEXT
2050 INK 0,13 :T(0)=13 :INK 2,26:T(2)=26
2060 INK 1,0:T(1)=0:INK 14,24:T(14)=24
2070 PAPER 1
2080 :
```

```
2090 ORIGIN 0,0,0,640,0,400
2100 MOVE 0,400:DRAWR 0,-335,1:DRAWR 639,0,1:DRAWR 0,334,1:DRAWR -639,0,1
2110 LOCATE 1,22:PEN 2:PRINT CHR$(244)+CHR$(245)+CHR$(246)+CHR$(247)+".";CHR$(252)+CHR$(253)+CHR$(254)+CHR$(255)+CHR$(242)
+CHR$(243) +" "+CHR$(248)+CHR$(249)+CHR$(250)+CHR$(251);T(CE)
2120 FOR I=0 TO 15 : PEN I:LOCATE I+2,23:PRINT CHR$(143):LOCATE I+2,24:PRINT CHR$(143):NEXT
2130 X=34:Y=48:FOR I=1 TO 16: MOVE X,Y:DRAWR 0,-32,1:X=X+32:NEXT
2140 LOCATE 2,25:PEN 1:FOR I=1 TO 16:PRINT CHR$(143);:NEXT:LOCATE 2,25 :PEN 2:PRINT CHR$(240)
2150 X=320:Y=200
2155 MOVE 543,0:DRAWR 0,64,1 :LOCATE 19,24:PRINT"?";
2160 ORIGIN 0,0,0,640,64,400:X=320:Y=200
2170 RETURN
2180 '
3000 '--------------------------------
3010 '  DEPLACEMENT DU CURSEUR  PALETTE
3020 '--------------------------------
3030 EXL=L
3040 IF INKEY(31)=0  AND L<17 THEN L=L+1:LOCATE EXL,25:PRINT " ";: LOCATE L,25:PEN (L-2):PRINT CHR$(240);: LOCATE 17,22:
PRINT T(L-2)
3050 IF INKEY(39)=0  AND L>2 THEN L=L-1:LOCATE EXL,25:PRINT " ";: LOCATE L,25:PEN (L-2):PRINT CHR$(240);: LOCATE 17,22:
PRINT T(L-2)
3055 CE=L-2
3060 IF C=0 THEN PEN(E) ELSE PEN CE
3070 LOCATE 5,22:IF P=3 THEN PRINT "." ELSE PRINT CHR$(143)
3080 IF INKEY(31)<>0 AND INKEY(39)<>0 THEN RETURN ELSE FOR I=1 TO 200:NEXT: GOTO 3030
3090 '
4000 '--------------------------------
4010 ' MODIFICATION DE LA PALETTE
4020 '--------------------------------
4030 INK CE,N:T(CE)=N
4040 LOCATE 17,22:PEN CE:PRINT T(CE)
4050 N=N+1:IF N=27 THEN N=0
4060 FOR I=1 TO 200:NEXT
4070 RETURN
5000 '--------------------------------
5010 '       DEPLACEMENT PINCEAU
5020 '   ET TESTS DES TOUCHES FONCTIONS
5030 '---------------------------------
5040 E=TEST(X,Y):IF C THEN 5060
5050 P=3:LOCATE 5,22:PEN(E):PRINT "."
5060 PLOT X,Y,2:FOR I=1 TO 5:NEXT:PLOT X,Y,1:FOR I=1 TO 5:NEXT
5070 IF P=3 THEN PLOT X,Y,CE ELSE CO=INT(X/32)+1:LI=25-INT (Y/16):IF CO>=1 AND CO<=20 AND LI<=25 AND LI>=1 THEN LOCATE
CO,LI:PEN CE: PRINT CHR$(143)
5080 IF C=0 THEN PLOT X,Y,E
5090 IF INKEY(27)=0 THEN P=9-P:FOR I=1 TO 200: NEXT: LOCATE 5,22:PEN CE:IF P=3 THEN PRINT"." ELSE PRINT CHR$(143)
5100 IF INKEY(39)=0 OR INKEY(31)=0 THEN GOSUB 3030
5110 IF INKEY(18)=0 THEN GOSUB 4030
5120 EXX=X:EXY=Y
5130 IF INKEY(1)=32 AND C=0 AND X<640 THEN X=X+20
5140 IF INKEY(8)=32 AND C=0 AND X>0 THEN X=X-20
5150 IF INKEY(0)=32 AND C=0 THEN IF Y<400 THEN Y=Y+15
```



```
5160 IF INKEY(2)=32 AND C=0 THEN IF Y>66 THEN Y=Y-15
5170 X=X+P*((1 AND INKEY(1)=0)-(1 AND INKEY(8)=0)):IF X<0 OR X>640 THEN X=EXX
5180 Y=Y+P*(1 AND INKEY(0)=0)/2:IF Y>=400 OR Y<66 THEN Y=EXY
5190 Y=Y-P*(1 AND INKEY(2)=0)/1.5:IF Y>=400 OR Y<66 THEN Y=EXY
5195 IF LOUPE THEN GOSUB 11000
5200 IF INKEY(9)=0 THEN C=1-C:FOR I=1 TO 50:NEXT:IF C=1 THEN PEN CE:LOCATE 5,22:PRINT ".":LOCATE 6,22:PEN 2:PRINT" ACTIF"
ELSE PEN 2 :LOCATE 6,22:PRINT CHR$(252)+CHR$(253)+CHR$(254)+CHR$(255)+CHR$(242)+CHR$(243)
5210 IF INKEY(60)=32  OR INKEY(60)=128 THEN GOSUB 6010
5220 IF INKEY(62)=32 OR INKEY(62)=128 THEN GOSUB 7010
5230 IF INKEY(16)=32 OR INKEY(16)=128 THEN ERASE T:GOTO 1000
5240 IF INKEY(29)=0 THEN DR=1:DX=X:DY=Y:PLOT X,Y,CE:PEN CE:LOCATE 1,25:PRINT"*"
5250 IF INKEY(36)=0 AND DR=1 THEN DRAW DX,DY,CE:DR=0:LOCATE 1,25:PEN 0:PRINT CHR$(143)
5260  IF INKEY(62)=0 AND DR=1 THEN R=SQR((X-DX)^2+(Y-DY)^2):DEG:PLOT DX+R,DY,CE:FOR A=0 TO 360 STEP 10:DRAW R*COS(A)+DX,
R*SIN(A)+DY,CE:NEXT A:DR=0:PEN 0:LOCATE 1,25:PRINT CHR$(143)
5270 IF INKEY(53)=0 THEN GOSUB 10000
5275 IF INKEY(54)=0 THEN BO=BO+1:FOR I=1 TO 100:NEXT:IF BO=27 THEN BO=0
5276 BORDER BO
5277 IF INKEY(44)=0 THEN GOSUB 12000
5280 IF INKEY(50)=0 THEN GOSUB 8010
5285 IF INKEY(30)=0 THEN FOR I=1 TO 100:NEXT:LOUPE=1-LOUPE:IF LOUPE=0 THEN CLS #3:LOCATE 19,24:PEN 2:PRINT"?";
5290 GOTO 5040
6000 '---------------------------------
6010 ' SAUVEGARDE   TABLEAU
6020 '---------------------------------
6030 WINDOW SWAP 0,1
6040 CLS:PEN 2:SPEED KEY 10,5
6050 IF INKEY$<>"" THEN 6050
6060 INPUT"NOM ?",N$
6070 SPEED WRITE 1
6080 SAVE N$,B,&C000,&4000
6090 CLS
6100 WINDOW SWAP 0,1
6110 GOSUB 2090
6130 SPEED KEY 8,1
6140 RETURN
7000 '---------------------------------
7010 '    CHARGEMENT  TABLEAU
7020 '---------------------------------
7030 WINDOW SWAP 0,1
7040 PEN 2
7050 CLS
7060 LOAD ""
7070 CLS
7080 WINDOW SWAP 0,1
7090 GOSUB 2090
7100 RETURN
8000 '---------------------------------
8010 '    S/PGM   FILL
8020 '---------------------------------
8030 DX=X:DY=Y:F=E
```



```
8040 DX=DX+3:IF TEST(DX,DY)=F AND DX<640 THEN 8040
8050 ZX=DX-X-6:DX=X
8060 DX=DX-3:IF TEST(DX,DY)=F AND DX>0 THEN 8060
8070 PLOTR 3,0,CE:DRAWR (X-DX)+ZX,0,CE
8080 DY=DY-2:DX=X
8090 IF INKEY(39)=0 OR INKEY(31)=0 THEN GOSUB 3030
8100 IF TEST(DX,DY)=F AND DY>64 AND INKEY(63)<>0 THEN 8040
8110 X=DX:Y=DY:PRINT"";:RETURN
9000 '--------------------------------
9010 '       PRESENTATION
9020 '--------------------------------
9030 SYMBOL AFTER 64
9040 SYMBOL 125,&60,&10,&3C,&66,&7E,&60,&3C
9050 SYMBOL 123,&6,&8,&3C,&66,&7E,&60,&3C
9060 SYMBOL 64,&60,&10,&78,&C,&7C,&CC,&76
9070 SYMBOL 92,&0,&0,&3C,&66,&60,&3E,&8,&18
9080 MODE 1:TR$=CHR$(22)+CHR$(1):NTR$=CHR$(22)+CHR$(0)
9090 INK 0,13:INK 1,1:INK 2,0:INK 3,11
9100 BORDER 13:PAPER 0
9110 PEN 3:FOR I=1 TO 3: PRINT TAB(13) STRING$(13,CHR$(143)) :NEXT
9120 PEN 1:PRINT TR$:LOCATE 15,2:PRINT"CPC PAINT":PRINT NTR$
9130 MOVE 200,350:DRAWR 200,0,2:DRAWR 0,40,2
9140 PEN 1:PRINT:PRINT TAB(30)"M. Chanaud"
9150 PEN 2:LOCATE 1,8:PRINT"Ce programme permet de cr"+CHR$(123)+"er "+CHR$(64)+" l'"+CHR$(123)+"cran"
9160 PRINT:PRINT"des images multicolores en d"+CHR$(123)+"pla"+CHR$(92)+"ant un":PRINT:PRINT"pinceau "+CHR$(64)+" l'aide des touches fl"+CHR$(123)+"ch"+CHR$(123)+"es,"
9170 PRINT:PRINT"de tracer des cercles et des segments,"
9180 PRINT:PRINT"de choisir sa palette de couleurs et de":PRINT:PRINT"la modifier "+CHR$(64)+" volont"+CHR$(123)+"."
9190 GOSUB 9400
9192 FOR I%=1 TO 3:PEN 3:PRINT STRING$(40,143);:NEXT
9200 PRINT TR$:PEN 2:LOCATE 1,1:PRINT CHR$(150)+STRING$(38,CHR$(154))+CHR$(156)+CHR$(149)+"   Les commandes disponibles sont:"+C HR$(149)+CHR$(147)+STRING$(38,CHR$(154))+CHR$(153)  :PRINT NTR$
9210 PRINT CHR$(154)+" FLECHES = D"+CHR$(123)+"placement du pinceau":PRINT:PRINT"  -------   SHIFT"+CHR$(123)+"es: avance rapide"
9220 PRINT:PRINT CHR$(154)+" COPY    = Pinceau TRANSPARENT / ACTIF"
9230 PRINT:PRINT CHR$(154)+" "+CHR$(199)+" ET "+CHR$(197)+"  = Couleur de la peinture"
9240 INK 1,26:PRINT:PRINT CHR$(154)+"    P     = Grosseur du pinceau ";:PEN 1:PRINT CHR$(132);:PEN 2:PRINT" ou ";:PEN 1:PRINT STRING$ (2,CHR$(143)):PEN 2
9250 PRINT:PRINT CHR$(154)+" ENTER   = Modification de la palette"
9260 PRINT:PRINT CHR$(154)+" CTRL  S = Sauvegarde d'une image"
9270 PRINT:PRINT CHR$(154)+" CTRL  C = Chargement d'une image"
9280 PRINT:PRINT CHR$(154)+" CTRL CLR= Effacement du tableau"
9290 FOR I=1 TO 100:NEXT:GOSUB 9400
9300 CLS
9310 PRINT:PRINT CHR$(154)+"  *   = Validation d'un point de d"+CHR$(123)+"part":PRINT" (extremit"+CHR$(123)+" ou centre)"
9320 PRINT:PRINT TAB(8)CHR$(159)+" C : Trac"+CHR$(123)+" d'un Cercle"
9330 PRINT:PRINT  TAB(8)CHR$(159)+" L : Trac"+CHR$(123)+" d'une Ligne"
9340 PRINT:PRINT:PRINT CHR$(154)+" F   = FILL (en mode transparent)"
9350 PRINT:PRINT CHR$(154)+"  R   = REMPLISSAGE  --- # ---"
```



```
9360 PRINT:PRINT CHR$(154)+"  X   = Sortie du FILL en cours"
9365 PRINT:PRINT CHR$(154)+"  ?   = LOUPE  (ou non)"
9366 PRINT:PRINT CHR$(154)+"  B   = Couleur de Bordure"
9367 INK 3,24:PEN 3:PRINT:PRINT CHR$(154)+"  H   = HELP !!":PEN 1
9370 FOR I=1 TO 100:NEXT:GOSUB 9400
9380 RETURN
9390 PEN 2
9400 IF INKEY$<>"" THEN 9400
9410 PEN 2:LOCATE 16,25:PRINT  CHR$(243)+" APPUYEZ SUR UNE TOUCHE":CALL &BB18:CLS:RETURN
9440 'DRAWR 30,0,2:DRAWR 0,30,2:RETURN
9460 '---------------------------
9461 '   ARRET DEMANDE ??
9462 '---------------------------
9470 CLS #1:PRINT #1,"voulez-vous arreter":PRINT #1:PRINT #1," "O/N§"
9480 WHILE INKEY$ <>"":WEND
9490 r$=UPPER$(INKEY$):IF r$="" THEN 9490
9500 IF r$="N" THEN GOSUB 2090:RETURN
9510 IF r$="O" THEN CLS #1:PRINT #1,"Voulez-vous sauver":PRINT#1,
"ce dessin sur une":PRINT #1,"cassette ? (O/N)"
9512 WHILE INKEY$ <>"":WEND
9513 r$=UPPER$(INKEY$):IF r$="" THEN 9513
9514 IF r$="O" THEN GOSUB 6000
9515 CLS #1:CALL &BB18:END
9520 GOTO 9480
10000 '---------------------------
10010 '       FILL    "FIN"
10030 '---------------------------
10040 XS%=X:YS%=Y
10050 DIM XP%(200),YP%(200)
10060 HEAD%=0:TAIL%=0
10070 GOSUB 10170
10080 TAIL%=(TAIL%+1) MOD 200
10090 XT%=XP%(TAIL%)
10100 YT%=YP%(TAIL%)
10110 IF TEST(XT%+4,YT%)=E THEN XS%=XT%+4:YS%=YT%:GOSUB 10170
10120 IF TEST(XT%-4,YT%)=E THEN XS%=XT%-4:YS%=YT%:GOSUB 10170
10130 IF TEST(XT%,YT%+2)=E THEN XS%=XT%:YS%=YT%+2:GOSUB 10170
10140 IF TEST(XT%,YT%-2)=E THEN XS%=XT%:YS%=YT%-2:GOSUB 10170
10150 IF HEAD%=TAIL%+1 THEN ERASE XP%,YP%:PRINT"";:RETURN
10160 IF INKEY(63)=0 THEN ERASE XP%,YP%:PRINT"";:RETURN ELSE
GOTO 10080
10170 PLOT XS%,YS%,CE
10180 HEAD%=(HEAD%+1) MOD 200
10190 XP%(HEAD%)=XS%:YP%(HEAD%)=YS%
10200 RETURN
10210 FOR KK=1 TO 10:NEXT:RETURN
11000 '-------------------------------
11010 '           LOUPE
11020 '-------------------------------
11030 LOCATE 18,23:PEN (TEST(X-4,Y+2)):PRINT CHR$(233);
11040 LOCATE 19,23:PEN (TEST(X,Y+2)):PRINT CHR$(233);
11050 LOCATE 20,23:PEN (TEST(X+4,Y+2)):PRINT CHR$(233);
11060 LOCATE 18,24:PEN (TEST(X-4,Y)):PRINT CHR$(233);
11070 LOCATE 19,24:PEN (TEST(X,Y)):PRINT CHR$(232);
11080 LOCATE 20,24:PEN (TEST(X+4,Y)):PRINT CHR$(233);
11090 LOCATE 18,25:PEN (TEST(X-4,Y-2)):PRINT CHR$(233);
11100 LOCATE 19,25:PEN (TEST(X,Y-2)):PRINT CHR$(233);
11110 LOCATE 20,25:PEN (TEST(X+4,Y-2)):PRINT CHR$(233);
11120 RETURN
12000 REM ################################# HELP
12010 RESTORE 12110
12020 A$=INKEY$:IF A$<>"" THEN 12020
12030 CLS #1
12040 FOR I%=1 TO 16:READ W$
12050 PRINT #1:PRINT #1,W$:PRINT #1
12060 FOR j=1 TO 200:NEXT
12070 CALL &BB18:NEXT
12080 CLS #1:GOSUB 2090
12090 RETURN
12100 REM DATAS DE MODE D'EMPLOI
12110 DATA FLECHES= Deplacement
12120 DATA FLECHES+SHIFT Rapide
12130 DATA COPY = Transpa/Actif
12140 DATA < > = Choix peinture
12150 DATA P = Grosseur pinceau
12160 DATA ENTER=Modif. palette
12170 DATA CTRL  S = Sauvegarde
12180 DATA CTRL  C = Chargement
12190 DATA CTRL CLR= Effacement
12200 DATA   *  = Pointage
12210 DATA   C  = Cercle
12220 DATA   L  = Ligne
12230 DATA   F  = Fill
12240 DATA   R  = Remplissage
12250 DATA   X  = Exit fill
12260 DATA   B  = Bordure
```

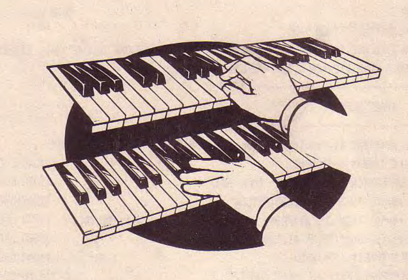
Extension RSX/MUSIC

L'Amstrad est équipé d'un générateur de son, programmable, assez sophistiqué, le AY-3-8912 de General Instruments. Celui-ci dispose de 3 canaux sur chacun desquels il est possible de programmer un signal de période et d'amplitude données.

Le Basic de l'Amstrad dispose d'une seule instruction, SOUND, pour gérer le PSG (Programmable Sound Register). Celle-ci offre de nombreuses possibilités, mais ne permet pas de créer directement une note de musique, dans une octave donnée. SOUND réclame impérativement la période correspondante. Le programme assembleur ci-dessous pallie cette carence: il complète le Basic par l'instruction I MUSIC qui demande quatre arguments:

- Le ou les canaux sur lesquels doit être produite la note désirée est un entier compris entre 0 et 7: pour envoyer la note sur l'un des canaux A,B ou C du PSG, il convient de mettre à 1 le bit correspondant (respectivement bits 0,1 et 2).
- L'octave est un entier compris entre 0 et 7 (de l'octave la plus grave à l'octave la plus aïgue).
- La note est un entier compris entre 0 et 11 (de sorte que les 12 demi-tons de chaque octave sont atteints).
- Le volume est un entier compris entre 0 et 15.

Cette nouvelle instruction se présente donc sous la forme: MUSIC, canaux, octave, note, volume.



Elle est aisément ajoutée au Basic, de l'Amstrad grâce à la souplesse de la ROM Basic et du système d'exploitation, qui offrent la possibilité de définir ses propres instructions sous la forme d'extensions du Basic, dites extensions RSX (pour Resident System Extensions). Le listing assembleur indique justement la marche à suivre pour créer une telle extension. Il est nécessaire de protéger la zone mémoire occupée par l'instruction en éxécutant un MEMORY &8FFF, et d'initialiser la nouvelle instrumention par un CALL &9000. Dès lors, le Basic reconnaîtra I MUSIC comme une instruction à part entière.

Il est également possible d'utiliser cette instruction dans un programme en langage machine. Les quatre arguments sont alors transmis dans les registres B,C,D et E du Z-80, et il convient d'appeler la routine MUSIC (se reporter au listing).

L'instruction fait également appel à la routine LECTUR qui permet la lecture des registres du PSG, et qui donne un exemple de programmation du port parallèle 8255 qui assure la liaison entre le circuit sonore et le microprocesseur. Attention avant de taper le programme, remplacer les "ù MUSIC" par "IMUSIC" et les £ par #.

Bernard Auré

```
Hisoft GENA3 Assembler. Page     1.

Pass 1 errors: 00

              ;  ****************************************************
              ;  **   Implememtation du BASIC AMSTRAD CPC 464/664   **
              ;  **     par addition de l'extension RSX  ùMUSIC:    **
              ;  **          MUSIC,canaux,octave,note,volume        **
              ;  **   ----------------------------------------     **
              ;  **          AURE Bernard / juillet 1985            **
              ;  ****************************************************
              ;
              ;
BD34          SNDREG EQU  £BD34      ;Routine MC SOUND REGISTER:
              ;                       Envoie la donnee contenue dans C
              ;                       dans le registre du PSG dont le
              ;                       numero est dans A.Preserve DE,HL.
BCD1          LOGEXT EQU  £BCD1      ;Routine KL LOG EXT:
```



```
                    150 ;                                 Ajoute une extension a la liste
                    160 ;                                 des extensions RSX. BC contient
                    170 ;                                 l'adresse de la table de commande
                    180 ;                                 de l'extension. HL pointe une
                    190 ;                                 zone libre de 4 octets en RAM.
                    200 ;
                    210 ;
9000                220          ORG   £9000
9000   010D90       230 INIT     LD    BC,TABRSX    ;Initialisation de
9003   210990       240          LD    HL,QUATRE    ;l'extension RSX ùMUSIC.
9006   C3D1BC       250          JP    LOGEXT
                    260 ;
9009                270 QUATRE   DEFS  4            ;Zone de 4 octets reservee a la
                    280 ;                           gestion de l'extension ùMUSIC.
900D   1290         290 TABRSX   DEFW  NOM          ;Table de commande de
900F   C31890       300          JP    MUSIC        ;l'extension RSX.
9012   4D555349     310 NOM      DEFM  "MUSI"       ;Nom de l'extension: le bit 7 du
9016   C3           320          DEFB  "C"+£80      ;dernier caractere est mis a 1.
9017   00           330          DEFB  0            ;Le 0 indique la fin de la table.
                    340 ;
9018   FE04         350 MUSIC    CP    4            ;Extension ùMUSIC proprement dite:
901A   C0           360          RET   NZ           ;Teste si nbre parametres=4.
901B   DD7E06       370          LD    A,(IX+6)     ;Range le type de canal (0 a 7)
901E   E607         380          AND   7            ;dans le
9020   57           390          LD    D,A          ;registre D.
9021   DD7E04       400          LD    A,(IX+4)     ;Range l'octave (0 a 7)
9024   E607         410          AND   7            ;dans le
9026   47           420          LD    B,A          ;registre B.
9027   DD7E02       430          LD    A,(IX+2)     ;Range la note (0 a 11)
902A   E60F         440          AND   15           ;dans le
902C   4F           450          LD    C,A          ;registre C.
902D   DD7E00       460          LD    A,(IX+0)     ;Range le volume (0 a 15)
9030   E60F         470          AND   15           ;dans le
9032   5F           480          LD    E,A          ;registre E.
9033   C34E90       490          JP    MUSIC0       ;Execution de la note, et fin.
                    500 ;
                    510 ;
                    520 ;TABLE des 12 periodes definissant les notes de l'octave 0
9036   EE0E         530 TBNOTE   DEFW  3822         ;00:DO
9038   180E         540          DEFW  3608         ;01:DO diese
903A   4D0D         550          DEFW  3405         ;02:RE
903C   8E0C         560          DEFW  3214         ;03:MI bemol




Hisoft GENA3 Assembler. Page     2.


903E   DA0B         570          DEFW  3034         ;04:MI
9040   2F0B         580          DEFW  2863         ;05:FA
9042   8F0A         590          DEFW  2703         ;06:FA diese
9044   F709         600          DEFW  2551         ;07:SOL
9046   6809         610          DEFW  2408         ;08:SOL diese
9048   E108         620          DEFW  2273         ;09:LA
904A   6108         630          DEFW  2145         ;10:SI bemol
904C   E907         640          DEFW  2025         ;11:SI
                    650 ;
                    660 ;
                    670 ;ENVOIE la note C(0 a 11) de l'octave B sur les
```

```
                 ;canaux musicaux indiques par le registre D,
                 ;avec le volume E(0 a 15).
                 ;
904E  79         MUSICO LD   A,C
904F  FE0C       LOOP00 CP   12          ;Teste si la note
9051  3805              JR   C,JMP00     ;est bien comprise
9053  D60B              SUB  11          ;entre 0 et 11.
9055  04                INC  B
9056  18F7              JR   LOOP00
9058  07         JMP00  RLCA
9059  213690            LD   HL,TBNOTE   ;Charge
905C  85                ADD  A,L         ;dans HL
905D  6F                LD   L,A         ;la
905E  3001              JR   NC,JMP01    ;periode
9060  24                INC  H           ;correspondant
9061  7E         JMP01  LD   A,(HL)      ;a
9062  23                INC  HL          ;l'octave
9063  66                LD   H,(HL)      ;zero.
9064  6F                LD   L,A
9065  05         LOOP01 DEC  B           ;Divise
9066  FA6F90            JP   M,JMP02     ;la periode
9069  CB3C              SRL  H           ;HL de base
906B  CB1D              RR   L           ;par
906D  18F6              JR   LOOP01      ;2^octave.
906F  7A         JMP02  LD   A,D
9070  2F                CPL
9071  4F                LD   C,A
9072  C5                PUSH BC
9073  3E07              LD   A,7         ;Lecture du registre 7 du PSG:
9075  CDA590            CALL LECTUR      ;contenu retourne dans A.
9078  C1                POP  BC
9079  A1                AND  C           ;Ouvre les canaux sur lesquels
907A  4F                LD   C,A         ;la note doit etre envoyee
907B  3E07              LD   A,7         ;(laisse invariant l'etat des
907D  CD34BD            CALL SNDREG      ; canaux non concernes).
9080  0600              LD   B,0         ;Envoie la periode HL et le volume
9082  CB3A       LOOP02 SRL  D           ;E sur les canaux indiques par les
9084  3018              JR   NC,JMP03    ;bits 0,1,2 du registre D:
9086  C5                PUSH BC          ;  Canal A=bit 0,Canal B=bit 1,
9087  78                LD   A,B         ;  Canal C=bit 2.
9088  07                RLCA
9089  F5                PUSH AF
908A  4D                LD   C,L         ;Envoie octet de poids faible
908B  CD34BD            CALL SNDREG      ;de la periode HL = L.
908E  F1                POP  AF
908F  3C                INC  A
9090  4C                LD   C,H         ;Envoie octet de poids fort


Hisoft GENA3 Assembler. Page     3.


9091  CD34BD            CALL SNDREG      ;de la periode HL = H.
9094  C1                POP  BC
9095  C5                PUSH BC
9096  3E08              LD   A,8
9098  80                ADD  A,B
9099  4B                LD   C,E         ;Envoie le
909A  CD34BD            CALL SNDREG      ;volume E.
```



```
909D  C1                   POP   BC
909E  04          JMP03    INC   B
909F  78                   LD    A,B
90A0  FE03                 CP    3
90A2  20DE                 JR    NZ,LOOP02
90A4  C9                   RET
                  ;
                  ;
                  ;LECTURE du registre du PSG dont le numero est dans A:
                  ;Le contenu du registre est retourne dans A,
                  ;Les registres DE et HL sont preserves.
                  ;Adresse du port A du 8255 -------------: £F400,
                  ;Adresse du port C du 8255 -------------: £F600,
                  ;Adresse du registre de commande du 8255: £F700.
                  ;
90A5  F3          LECTUR   DI
90A6  06F4                 LD    B,£F4      ;Envoie numero du registre du
90A8  ED79                 OUT   (C),A      ;PSG sur le port A du 8255.
90AA  06F6                 LD    B,£F6
90AC  ED78                 IN    A,(C)      ;Selectionne ce registre
90AE  F6C0                 OR    £C0        ;(bits 6 & 7 du port C
90B0  ED79                 OUT   (C),A      ; du 8255 mis a 1).
90B2  E63F                 AND   £3F        ;Retour en mode inactif
90B4  ED79                 OUT   (C),A      ;(bits 6 & 7 remis a 0).
90B6  4F                   LD    C,A
90B7  04                   INC   B          ;Programme le port A du 8255
90B8  3E92                 LD    A,£92      ;en entree (bit 4 du registre
90BA  ED79                 OUT   (C),A      ; de commande mis a 1).
90BC  79                   LD    A,C
90BD  F640                 OR    £40        ;Mode lecture des registres du PSG
90BF  05                   DEC   B          ;(bit 6 du port C a 1,
90C0  ED79                 OUT   (C),A      ; bit 7 du port C a 0).
90C2  06F4                 LD    B,£F4      ;Lecture du contenu du registre
90C4  ED78                 IN    A,(C)      ;du PSG selectionne.
90C6  F5                   PUSH  AF
90C7  06F7                 LD    B,£F7      ;Reprogramme le port A du 8255
90C9  3E82                 LD    A,£82      ;en sortie (bit 4 du registre
90CB  ED79                 OUT   (C),A      ; de commande remis a 0).
90CD  05                   DEC   B          ;Retour en mode inactif
90CE  ED49                 OUT   (C),C      ;(bits 6 & 7 du port C remis a 0).
90D0  F1                   POP   AF
90D1  FB                   EI
90D2  C9                   RET
                  ;

Pass 2 errors: 00

INIT    9000  JMP00   9058  JMP01   9061  JMP02   906F
JMP03   909E  LECTUR  90A5  LOGEXT  BCD1  LOOP00  904F
LOOP01  9065  LOOP02  9082  MUSIC   9018  MUSIC0  904E
NOM     9012  QUATRE  9009  SNDREG  BD34  TABRSX  900D


Hisoft GENA3 Assembler. Page     4.


TBNOTE 9036

Table used:   224 from  1000
```

Gestion de fichier

Description:

Ce mois-ci nous vous proposons un programme de gestion de fichier assez puissant puisqu'il possède de nombreuses options telles que la recherche de fiches par un mot, ou une phrase; et qu'il vous permet de créer des format de fiches de 8 critères, chaque critère comprenant jusqu'à 256 caractères. Précisons également que ce programme est compatible avec les disquette ainsi qu'avec les cassettes, l'adaptation étant automatique selon la configuration de votre AMSTRAD.

Fonctionnement:

Une fois, le programme entré, et le lancement effectué, vous voyez apparaître à l'écran un menu avec toutes les grandes familles d'opérations possibles (édition, sauvegarde, etc.).
Indiquez votre choix, vous n'avez plus qu'à suivre les indications fournies dans le programme.
Note: en mode recherche par défilement, vous pouvez interrompre celui-ci par une action sur la touche "escape" et revenir au menu en appuyant sur la touche S.

E. Charton

```
10 REM  ££££££££££££££££££££££££££££££££££££££
110 REM ££                                  ££
210 REM ££          gestion de fichier      ££
310 REM ££  copyright e.charton et AMSTRAD  ££
410 REM ££            magazine              ££
510 REM ££                                  ££
610 REM ££££££££££££££££££££££££££££££££££££££
710 REM
810 REM      --- initialisation ---
910 REM
1010 INK 0,0:INK 1,21:PEN 1:PAPER 0:BORDER 0
1110 REM
1210 REM       ---menu de presentation---
1310 REM  --et choix  des differentes  options --
1410 MODE 1
1510 LOCATE 10,3:PRINT "gestion de fichier"
1610 LOCATE 5,7:PRINT "1 recherche"
1710 LOCATE 5,8:PRINT "2 creation de fichier"
1810 LOCATE 5,9:PRINT "3 sauvegarde d'un fichier"
1910 LOCATE 5,10:PRINT "4 chargement d'un fichier"
2010 LOCATE 5,11:PRINT "5 edition d'un fichier"
2110 LOCATE 5,12:PRINT "6 sortie sur imprimante"
2210 REM ----------------------------------
2310 REM
2410 REM     acces au choix
2510 REM
2610 REM ----------------------------------
2710 a$=INKEY$:IF a$="" THEN 2710
2810 IF a$="1" THEN 9410
2910 IF a$="2" THEN 3510
3010 IF a$="3" THEN 16110
3110 IF a$="4" THEN 17010
3210 IF a$="5" THEN 11110
3310 IF a$="6" THEN 17910
3410 GOTO 2710
3510 REM ---------------------------------
3610 REM
3710 REM    creation de fiches
3810 REM
3910 REM ---------------------------------
4010 MODE 2 :CLEAR
```



```
4110 INPUT "nombre de criteres?(maxi 8)",crit:IF crit>8 THEN 4010
4210 DIM a$(100,crit):DIM n$(crit)
4310 FOR x=1 TO crit:PRINT "nom du critere ";x;" ":INPUT n$ (x):NEXT
4410 CLS
4510 FOR x=1 TO crit
4610 PRINT"-------------------------------------------------------------------
-----------"
4710 PRINT "--- CRITERE No:";X;"---     ";n$(x)
4810 NEXT
4910 PRINT"-------------------------------------------------------------------
-----------"
5010 PRINT"   ETES VOUS D'ACCORD? O/N":INPUT W$:IF W$="N" OR W$="n" THEN
     3510
5110 GOTO 1110
5210 REM ----------------------------------
5310 REM
5410 REM   defilement des fiches
5510 REM
5610 REM ----------------------------------
5710 MODE 2
5810 INPUT "par quelle fiche voulez vous commencer?",w
5910 IF w=0 THEN w=1
6010 FOR x=w TO 100
6110 PRINT "          fiche no:";x
6210 FOR y=1 TO crit
6310 IF INKEY(60)=0 THEN GOTO 1110
6410 IF INKEY(27)=0 THEN GOTO 6410
6510 GOSUB 7310
6610 NEXT y
6710 PRINT "------------------------------------------------------------------
-----------"
6810 PRINT" appuyez sur une touche pour continuer"
6910 IF INKEY$="" THEN 6910
7010 CLS
7110 NEXT x
7210 w=1:GOTO 6010
7310 REM ----------------------------------
7410 REM
7510 REM    sous routine d'affichage des
7610 REM               fiches
7710 REM
7810 REM ----------------------------------
7910 PRINT "------------------------------------------------------------------
-----------";n$(y);"  :";:PRINT a$(x,y)
8010 RETURN
8110 REM ----------------------------------
8210 REM
8310 REM  routine d'affichage d'une
8410 REM    fiche precise
8510 REM
8610 REM ----------------------------------
8710 MODE 2
8810 INPUT "quelle fiche desirez vous connaitre :",w
8910 IF w<1 OR w>100 THEN 8710
9010 x=w:FOR y=1 TO crit:GOSUB 7910:NEXT
9110 PRINT "------------------------------------------------------------------
--------"
9210 IF INKEY$="" THEN 9210
```

```
9310 GOTO 1110
9410 REM ---------------------------------
9510 REM
9610 REM   menu des recherches
9710 REM
9810 REM ---------------------------------
9910 MODE 1
10010 LOCATE 9,3:PRINT "recherche de fiches"
10110 LOCATE 3,5:PRINT "1 defilement des fiches"
10210 LOCATE 3,6:PRINT "2 recherche d'un numero"
10310 LOCATE 3,7:PRINT "3 recherche par mot"
10410 LOCATE 3,8:PRINT "4 retour"
10510 e$=INKEY$
10610 IF e$="1" THEN 5210
10710 IF e$="2" THEN 8110
10810 IF e$="3" THEN 20010
10910 IF e$="4" THEN 1110
11010 GOTO 10510
11110 REM ---------------------------------
11210 REM
11310 REM   menu de l'edition
11410 REM
11510 REM ---------------------------------
11610 MODE 1
11710 LOCATE 14,3:PRINT "edition"
11810 LOCATE 3,8:PRINT "1 entree de donnees"
11910 LOCATE 3,10:PRINT"2 correction de donnees"
12010 LOCATE 3,12:PRINT"3 retour"
12110 e$=INKEY$
12210 IF e$="1" THEN 12810
12310 IF e$="2" THEN 14410
12410 IF e$="3" THEN 1110
12510 GOTO 12110
12610 STOP
12710 MODE 2:LIST-
12810 REM ---------------------------------
12910 REM
13010 REM  entree de donnees
13110 REM
13210 REM ---------------------------------
13310 MODE 2:INPUT "par quelle fiche desirez vous commencer?",w
13410 CLS:IF w<0 OR w>100 THEN 13310
13510 PRINT "          fiche no:";w
13610 PRINT "-----------------------------------------------------------------------------
-------------"
13710 FOR x=1 TO crit
13810 PRINT n$(x);:INPUT a$(w,x)
13910 PRINT "-----------------------------------------------------------------------------
-------------"
14010 NEXT
14110 INPUT "un autre fiche o/n ",q$:IF q$="n" THEN 11110
14210 w=w+1:CLS:IF w>100 THEN w=1
14310 GOTO 13510
14410 REM ---------------------------------
14510 REM
14610 REM      correction des donnees
14710 REM
14810 REM ---------------------------------
```



```
14910 MODE 2:INPUT "quelle fiche desirez vous corriger?",w
15010 CLS :IF w<0 OR w>100 THEN 14910
15110 PRINT "           fiche no:";w
15210 PRINT "------------------------------------------------------------------------------
--------------"
15310 FOR x=1 TO crit
15410 PRINT n$(x);:PRINT a$(w,x)
15510 PRINT "------------------------------------------------------------------------------
--------------"
15610 PRINT "correction:";:INPUT a$(w,x)
15710 PRINT "------------------------------------------------------------------------------
--------------"
15810 NEXT:INPUT "un autre fiche o/n ",q$:IF q$="n" THEN 11110
15910 w=w+1:CLS:IF w>100 THEN w=1
16010 GOTO 15110
16110 REM ----------------------------------
16210 REM
16310 REM        sauvegarde
16410 REM
16510 REM ----------------------------------
16610 OPENOUT "fichier"
16710 PRINT £9,crit:FOR s=1 TO crit:PRINT £9,n$(s):NEXT s:FOR s=1 TO 100:
FOR e=1 TO crit:PRINT £9,a$(s,e):NEXT e:NEXT s
16810 CLOSEOUT
16910 GOTO 1110
17010 REM ----------------------------------
17110 REM
17210 REM        chargement
17310 REM
17410 REM ----------------------------------
17510 CLEAR:OPENIN "fichier"
17610 INPUT £9,crit:DIM a$(100,crit):DIM n$(crit):FOR s=1 TO crit:INPUT £9,
n$(s) :NEXT s:FOR s=1 TO 100:FOR e=1 TO crit:INPUT £9,a$(s
,e):NEXT e:NEXT s
17710 CLOSEIN
17810 GOTO 1110
17910 REM ----------------------------------
18010 REM
18110 REM      sortie sur imprimante
18210 REM
18310 REM ----------------------------------
18410 MODE 2
18510 LOCATE 30,3:PRINT "sortie sur imprimante"
18610 LOCATE 4,7
18710 INPUT "a partir de quelle fiche desirez vous effectuer la sortie :",q
18810 INPUT "longueur de la sortie (en nombre de fiches)",l
18910 PRINT:PRINT
19010 INPUT "OK o/n",y$:IF y$="n" THEN 18410
19110 FOR s=q TO l
19210 PRINT £8,"fiche no";s:GOSUB 19910
19310 FOR t=1 TO crit
19410 PRINT £8,n$(t);a$(s,t)
19510 GOSUB 19910:NEXT t
19610 PRINT £8:PRINT £8
19710 NEXT s
19810 GOTO 1110
19910 PRINT £8,"-------------------------------------------------------------------------------
--------------":RETURN
```

```
20010 REM -----------------------------------
20110 REM
20210 REM       recherche par mot
20310 REM
20410 REM -----------------------------------
20510 MODE 2
20610 LOCATE 27,3:PRINT" recherche par mot"
20710 LOCATE 2,6:INPUT "entrez le mot a traiter . : ";m$
20810 lo=LEN(M$)
20910 FOR f=1 TO 100
21010 FOR g=1 TO crit
21110 lol=LEN (a$(f,g))
21210 FOR d=1 TO lol:t$=MID$(a$(f,g),d,lo):LOCATE 4,8:PRINT t$:IF t$=m$
THEN GOTO 21510
21310 NEXT d
21410 NEXT g:NEXT f:CLS:PRINT"la recherche n'a pas aboutie":FOR f=1 TO
2000:NEXT:GOTO 1110
21510 PRINT "    le mot recherche a ete trouve dans la fiche no ";f;" ,
desirez vous": PRINT "          consulter cette fiche  (o/n)";:
INPUT c$:IF c$="o" THEN GOSUB 21810
21610 INPUT "    desirez vous poursuivre la recherche (o/n)",c$:IF c$="n"
THEN 1110
21710 GOTO 21310
21810 REM -----------------------------------
21910 REM
22010 REM  sous routine d'affichage de la
22110 REM          fiche trouvee
22210 REM
22310 REM -----------------------------------
22410 CLS
22510 PRINT "fiche no ",f:PRINT "--------------------------------------------
----------------------------------"
22610 FOR h=1 TO crit:PRINT n$(h);" :";a$(f,h):PRINT"---------------------
--------------------------------------------------------
":NEXT h
22710 IF INKEY$="" THEN 22710
22810 CLS:RETURN
```

Dessin d'histogrammes

Vous voulez réaliser de très beaux histogrammes en trois dimensions tout en tapant un très petit programme ? Alors essayez donc "Dessins d'histogrammes"... Il s'agit d'un programme de quatre-vingts lignes en Basic, réalisé sur Amstrad 664. Outre les histogrammes, l'intérêt de ce programme réside dans les bandes FOR NEXT imbriquées et les utilisations de mode graphique et texte dans le même écran.

Nicholas Seiersen

```
1 CLS
2 LOCATE 10,10:PRINT "DESSIN D'HISTOGRAMMES "
3 LOCATE 13,12:PRINT "EN 3 DIMENSIONS"
4 LOCATE 10,15:PRINT "PAR NICHOLAS SEIERSEN"
5 FOR X=1 TO 2000:NEXT
10 MODE 1:INK 2,3:INK 3,6
20 f$(0)= "A"
30 f$(1)= "B"
40 f$(2)= "C"
50 f$(3)= "D"
60 f$(4)= "E"
70 f$(5)= "F"
80 f$(6)= "G"
90 f$(7)= "H"
105 INPUT "TITRE DU GRAPH";t$
110 INPUT "QUELLE EST LA PREMIERE ANNEE";y
```



```
120 INPUT "COMBIEN DE FAMILLES DE PRODUITS (MAX 6)";n:IF n>6 THEN 120
130 n=n-1
140 FOR i=0 TO 4 'rem boucles de saisie
150 FOR j=0 TO n
160 PRINT "YEAR ";y+i;" FAMILLE ";f$(j):INPUT r(i,j)
170 IF r(i,j)>m THEN m=r(i,j)
180 NEXT j
190 PRINT
200 NEXT i
210 MODE 2
220 FOR i=0 TO 4  'rem affichage des donnees
230 FOR j=0 TO n
240 PRINT i;r(i,j);
250 NEXT j::
260 PRINT:NEXT i
270 PRINT "le maximum est";m;" l'echelle est de 1:";m/16;
280 PRINT "voulez vous resaisir? o/n"
290 x$=INKEY$
300 IF x$="o" OR x$="O" THEN 160:ELSE IF x$="" THEN 290
310 MODE 1 'rem dessin de la grille
311 MOVE 128,144:DRAW 48,64:DRAW 48,304:DRAW 128,384:DRAW 608,384:DRAW 608,144:DRAW 528,64:DRAW 48,64
313 MOVE 128,384:DRAW 128,144
314 MOVE 48,176:DRAW 128,260:DRAW 608,260
315 MOVE 144,64:DRAW 224,144:DRAW 224,384
316 MOVE 240,64:DRAW 320,144:DRAW 320,384
317 MOVE 336,64:DRAW 416,144:DRAW 416,384
318 MOVE 432,64:DRAW 512,144:DRAW 512,384
319 MOVE 128,144:DRAW 608,144
330 FOR j=n TO 0 STEP -1 'rem dessin de l'arrete de l'histogramme
340 FOR i=4 TO 0 STEP -1
350 PEN 3:PAPER 0
360 y=21-INT((r(i,j)/m)*16)-i
370 x=4+i+6*j
380 LOCATE x,y
390 PRINT CHR$(214) 'dernier carre en perspective-triangle
400 FOR c=y+1 TO 21-i
410 LOCATE x,c:PRINT CHR$(143) 'carre plein
420 NEXT c
430 NEXT i
440 NEXT j
460 PEN 2 'rem dessin des cotes de l'histogramme
470 FOR j=0 TO n
480 FOR i=0 TO 4
490 PAPER 3
500 x=5+i+6*j
510 y=21-INT((r(i,j)/m)*16)-i
520 LOCATE x,y:PRINT CHR$(214) 'dernier carre en perspective-triangle
530 PAPER 0
540 LOCATE x,(21-i):PRINT CHR$(212) 'premier carre en
perspective-triangle inverse
550 FOR c=y+1 TO 20-i
560 LOCATE x,c:PRINT CHR$(143)  'carre plein
570 NEXT c
580 NEXT i
590 NEXT j
595 PEN 1
610 FOR j=0 TO n   'affichage des familles
620 LOCATE 4+6*j,22
630 PRINT f$(j)
640 NEXT
650 LOCATE 1,6  'affichage de l'echelle
660 PRINT INT(m)
670 LOCATE 1,14:PRINT INT(m/2)
680 LOCATE 1,21
690 PRINT " 0 "
700 LOCATE 4,23:PRINT t$
```



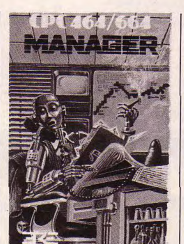
Création de US Gold France

A l'issue du Personnal Computer World Show qui s'est tenu à Londres du 4 au 8 septembre, Geoff Brown, Président de US Gold LTD et A. Loridan directeur de Micromania, DL Distribution ont annoncé qu'un accord de coopération avait été conclu pour la création immédiate de US Gold France.
La création de US Gold France répond à deux besoins :
Assurer une meilleure diffusion des produiis US Gold en France. US Gold publie en Angleterre les meilleurs titres développés par les sociétés américaines (Access, Data Soft, Microprose, Sega) sous une forme plus adaptée au marché européen.
Les jeux sont édités principalement sous forme de K7 au lieu de disquettes et leur prix varie de 95 à 130 F.
Ces produits étaient jusqu'à présent disponibles en France par l'intermédiaire de divers importateurs dans leur version anglaise.
US Gold France va dorénavant distribuer ces mêems produits avec une notice en français et assurer une promotion soutenue des nouveaux titres dans la presse spécialisée. D'autre part, l'introduction de nouveaux produits se fera de façon simultanée en France et en Angleterre. Les premiers titres à paraître sont pour Amstrad : Raid Over Moscou, Bruce Lee, Pole Position, Beach Head 2.
Les premiers jeux seront disponibles courant octobre et les introduciuons se feront ensuite régulièrement.
La création de US Gold France devrait contribuer au développement de la micro-informatique en France en élargissant la gamme de logiciels français de qualité, disponibles sur l'ensemble des micro-ordinateurs vendus en France.

US Gold France, B.P. 3, Zac des Mousquettes, 06740 Châteauneuf de Grasse. Tél. : (93) 42.57.12. Albert Loridan.

Manager, Gutter

Ere informatique sort, au moment du Sicob, deux nouveaux jeux pour le CPC 464 et le CPC 664.
Manager vous transforme en patron de choc dans un secteur que vous connaissez bien : la micro-informatique. Chaque trimestre, vos trois chefs de département viennent au rapport et attendent vos décisions. A vous de décider, vite et bien. Manager va immédiatement vous montrer les conséquences économiques de vos ordres et les répercuter à toutes les personnes du comité directeur. Vous devrez vous battre contre l'inflation, un coefficient de vétusté de votre entreprise, les aléas de la vie commerciale de vos produits. Le climat social risquera à tout instant de se détériorer, vous pourrez être obligé de licencier... (Chefs d'entreprises, essayez-vous donc à Manager...!)



Gutter est nettement moins lourd d'enseignement sur notre vie quotidienne : une gouttière de métal peuplée de figurines médiévales et d'objets magiques est dévalée par une bille d'acier. Vous devrez, en dirigeant la boule d'acier détruire certains personnages et en éviter d'autres afin de pouvoir recueillir les objets magiques.
Ere Informatique, 27, rue de Léningrad, 75008 PARIS.

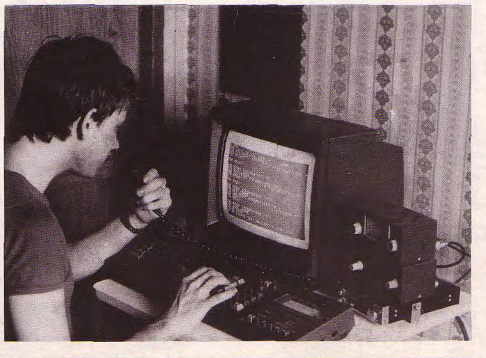


QRZ? AMSTRAD!

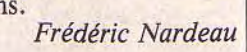
Heureux possesseurs d'un AMSTRAD flambant neuf, une fois le manuel d'utilisation lu, vous tombez dans les affres de la recherche d'une "âme sœur" avec qui partager trucs et astuces, livres et logiciels...

Au début, tout va bien. Le manuel fournit des exemples de programmes et puis... vous voilà bloqué. Une des solutions est de s'inscrire dans un club Informatique où vous pourrez certainement rencontrer — ô joie ! — d'autres Amstradistes pour vous faire expliquer tel ou tel point du manuel. Une autre est de passer une petite annonce du style "AMSTRAD solitaire cherche correspondant...". Mais voilà... la recherche est souvent difficile, le résultat parfois peu concluant. Le club n'est peut-être pas près de chez vous, le correspondant risque de se trouver à l'autre bout de la France et, vos diverses occupations terminées, il est peut-être trop tard pour aller discuter avec d'autres adeptes. Pourtant il existe une solution. Dans la banlieue sud de Paris, parfois très tard dans la nuit, se tiennent des discussions informatiques entre possesseurs d'AMSTRAD, les uns étant confortablement installés dans leur lit, les autres devant leur clavier. Leur astuce ? Une bande de fréquence radio, située sur le 27 mégahertz et appelée plus communément C.B. (Citizen Band). Ainsi, divers cibistes se réunissent sur les ondes, échangent leurs dernières trouvailles, prennent des contacts en vue d'échanges de livres, de revues, de logiciels, etc.

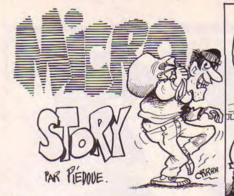
Il vous suffit pour cela d'un petit émetteur/récepteur C.B., d'une alimentation 12 volts, d'une petite antenne et d'une licence qu'on se procure en acquittant une taxe de 170 F (pour 5 ans) après des PTT. Vous n'avez donc pas, pour émettre, besoin de connaissances en radio. Et le tour est joué ! Vous pouvez vous occuper de votre AMSTRAD préféré même vers 23 heures, heure à laquelle il est indélicat d'aller sonner chez ses amis. Les cibistes, eux, se couchent tard. Décidez cette installation avec quelques-un de vos amis et vous pourrez vous livrer à vos conversations favorites à n'importe quelle heure du jour ou de la nuit. De toutes les manières, même si vous décidez seul d'acheter ce matériel C.B., il est fort probable que vous rencontrerez d'autres cibistes qui partagent le même intérêt que vous et qu'on peut entendre, de temps à autres, mettre au point un programme, par la voie des ondes, à quelques kilomètres de distance les uns des autres. Finie la solitude et bonjour les contacts ! Lancez-vous avec quelques amis et vous verrez que vous attirerez du monde à l'écoute de vos conversations informatiques. La C.B. est mieux adaptée pour cela que le téléphone car elle permet des réunions de plus de deux personnes (et elle ne coûte rien en communication !). Bien sûr, vous devrez parler à tour de rôle mais avec un peu d'autodiscipline, cela donne des résultats très concluants.



En bref, pour environ 1.000 F, vous pourrez prolonger vos conversations et rencontrer, chez vous, avec un peu de chance et de persévérance, un maximum d'utilisateurs qui pourront vous aider. En outre, cela vous permettra de vous découvrir une autre passion : celle de la radiocommunication. Si vous voulez plus de renseignements sur cette idée, sachez qu'il existe des revues spécialisées sur la C.B. De toutes façon, commencez par regarder autour de vous ; il se trouve sûrement un cibiste qui se fera un plaisir de vous renseigner et de vous aider à choisir votre matériel en fonction de vos besoins.



Frédéric Nardeau


MICRO
STORY
PAR PIÉDOUE.
CRRRR

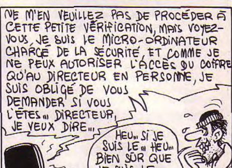

ÇA C'EST ENNUYEUX! JE VOULAIS JUSTEMENT CAMBRIOLER.
SI VOUS ÊTES UN VOLEUR, INTERDIT DE TOURNER CETTE POIGNÉE ET D'ENTRER


TANT PIS, JE TRANSGRESSE! JE SUIS COMME ÇA, MOI!
HÉ! HÉ!
BON! MAINTENANT, SE REPÉRER ET TROUVER LE COFFRE!


CLIC!
HEU... MERCI!
LE COFFRE DITES-VOUS? 3e PORTE À DROITE.


ILS SONT DINGUES DANS CETTE BOÎTE! ILS ME MÂCHENT LE BOULOT! ENFIN C'EST LEUR PROBLÈME!
3e PORTE.

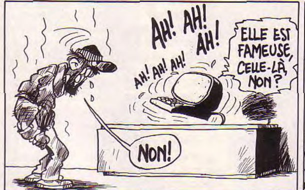

BONJOUR!.. C'EST VOUS QUI CHERCHEZ LE COFFRE?
HEU... C'EST-À-DIRE.. OUI...


NE M'EN VEUILLEZ PAS DE PROCÉDER À CETTE PETITE VÉRIFICATION, MAIS VOYEZ-VOUS, JE SUIS LE MICRO-ORDINATEUR CHARGÉ DE LA SÉCURITÉ, ET COMME JE NE PEUX AUTORISER L'ACCÈS DU COFFRE QU'AU DIRECTEUR EN PERSONNE, JE SUIS OBLIGÉ DE VOUS DEMANDER SI VOUS L'ÊTES... DIRECTEUR, JE VEUX DIRE...
HEU.. SI JE SUIS LE... HEU... BIEN SÛR QUE JE SUIS LE DIRECTEUR, HÉ!

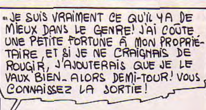

DANS CE CAS, IL FAUT AUSSI QUE VOUS PRONONCIEZ "BARAMU" CE QUI EST LE MOT DE PASSE PROUVANT VOTRE IDENTÉ...
BARAMU, HÉ!
L'IMBÉCILE!
BARAMU!
ÇA IRA, HÉ?


BRAVO! VOUS ÊTES TOMBÉ PILE! VOUS N'AVEZ PLUS QU'À ALLER APPUYER SUR CE MINUSCULE BOUTON ET LE COFFRE S'OUVRIRA...
C'EST PAS VRAI ?!
ET BIEN MOI, IL COMMENCE À ME PLAIRE, LE MODERNISME


OUAH!


AH! AH! AH!
AH! AH! AH!
ELLE EST FAMEUSE, CELLE-LÀ, NON?
NON!


ET BIEN, IL FAUDRA VOUS Y FAIRE! VOUS AVEZ DEVANT VOUS LE SUPER-MAXI-MICRO-ORDINATEUR FACÉTIEUX ET INFAILLIBLE. LÀ OÙ JE SUIS INSTALLÉ, ON NE PEUT RIEN VOLER.
HA! OUI?
OUI!..


...JE SUIS VRAIMENT CE QU'IL Y A DE MIEUX DANS LE GENRE! J'AI COÛTÉ UNE PETITE FORTUNE À MON PROPRIÉTAIRE, ET SI JE NE CRAIGNAIS DE ROUGIR, J'AJOUTERAIS QUE JE LE VAUX BIEN. ALORS DEMI-TOUR! VOUS CONNAISSEZ LA SORTIE!


CETTE COCHONNERIE A RAISON. JE SUIS COINCÉ!

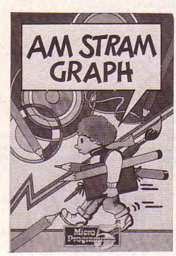

BAH!


MAIS?


UNE PETITE FORTUNE, IL A DIT?


AU SECOURS!.. AU RAPT!.. A L'ERREUR DE PROGRAMMATION!
RIEN VOLER, HEIN?..
?

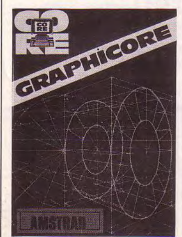
AMS TRAM GRAPH



Éditeur : *Micro programme 5*
Support : *cassette*
Genre : *utilitaire graphique*
Graphisme : ★★
Intérêt : ★★
Difficulté : ★★★
Appréciation : ★★

Ce programme multifonctions (utilitaire de création graphique, éditeur de matrice) intègre de nombreuses fonctions telles que le déplacement de l'écran dans une zone mémoire, tracé de triangles, tracé de cercles, tracés de boules, tracé de rectangles, scrolling vers le haut ou le bas.
Pourtant l'absence de menu dans le programme (largement remplacé par un manuel bien écrit, il est vrai), le manque de fonction habituellement présent dans les logiciels graphiques, tel que Fill (remplissage de zone), ou bien le zoom qui permet d'agrandir une partie de l'écran), se fait cruellement sentir. De plus la lenteur des déplacements, sans doute due au basic, pose quelques problèmes.
Amstramgraph est un programme, qui séduira les apprentis graphistes désireux de se familiariser avec ce type de logiciel.

GRAPHICORE



Éditeur : *Core*
Support : *cassette*
Genre : *utilitaire graphique*
Graphisme : ★★★★★
Intérêt : ★★★★
Difficulté : ★
Appréciation : ★★★★

Ce superbe utilitaire graphique n'a qu'un seul défaut : celui de ne fonctionner qu'en mode 2. La présentation d'ensemble du logiciel (menu pleine page facilement accessible, écran bien disposé), fait tout de suite penser à un programme professionnel. De plus la simplicité d'emploi est déroutante : à tout moment vous pouvez accéder au menu détaillant chacune des fonctions, et indiquant pour chacune, la lettre correspondante, il ne vous reste plus qu'à taper sur la touche correspondante pour obtenir la fonction désirée.
Des fonctions de quadrillage de l'écran, tracé de cercle, affichage de texte au curseur courant, reproduction d'une partie de dessin en n'importe quelle place sur l'écran, sont présentes. De plus une option "accès à d'autres programmes" laisse présager que ce logiciel sera évolutif et que l'on pourra acquérir d'autres fonctions. Quelques défauts tout de même, un Fill peu puissant et le manque d'option zoom. Un excellent logiciel tirant parti au maximum des capacités de l'AMSTRAD.

AMSPRITE



Éditeur : *The Electric Studio*
Support : *cassette*
Genre : *utilitaire de programmation*
Graphisme : ★★★
Intérêt : ★★★★
Difficulté : ★★★
Appréciation : ★★★★

Malgré ses nombreuses qualités, l'AMSTRAD est une machine qui ne possède pas de Sprites en version de base comme le 664 ou les MSX. Voici pour remédier à ce problème, qui n'en est plus un, une superbe routine de Sprites, extrêmement performante. L'ensemble est présenté sous la forme d'un pack de 2 cassettes, la première contenant un éditeur, la seconde, la routine en langage machine servant à gérer les Sprites ainsi qu'une courte démonstration sans grand intérêt d'ailleurs, mais démontrant assez bien les possibilités de l'ensemble.
Les Sprites, en couleur, peuvent prendre une taille variable définie par l'utilisateur. Le manuel très détaillé fournit un grand nombre de renseignements très précieux sur la routine. Un excellent outil de création pour les passionnés.

SEMDRAW

Éditeur : *Sémaphore*
Support : *cassette*
Genre : *utilitaire graphique*
Graphisme : ★★★
Intérêt : ★★★★
Difficulté : ★
Appréciation : ★★★★

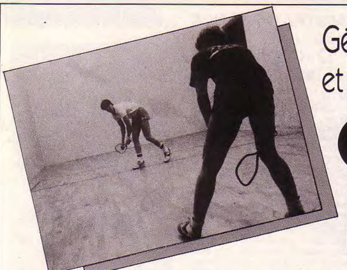
Cet utilitaire même s'il n'apporte rien de franchement nouveau dans sa catégorie n'en reste pas moins d'excellente qualité. La présentation est claire, et permet grâce à des instructions très simples d'en maîtriser rapidement toutes les subtilités. Le manuel très détaillé fourni avec la cassette explique clairement chaque mode.
Une démo très complète, accessible par le menu vient compléter le tout, même si le dessin fourni n'est pas d'une grande qualité, il démontre assez bien toutes les possibilités de SEMDRAW.
Le logiciel possède également des défilements d'écran et des effets de couleur.

MATRIX



Éditeur : *Logi'stick*
Support : *cassette*
Genre : *utilitaire graphique*
Graphisme : ★★★
Intérêt : ★★★
Difficulté : ★★
Appréciation : ★★★

Matrix est un utilitaire graphique très complet permettant de travailler les matrices de caractères de l'AMSTRAD. Il comprend de nombreuses options d'édition et de dessin, permettant d'effectuer un maximum d'opérations sur les caractères redéfinissables.
L'écran très complet comprend un quadrillage dans lequel on édite la matrice, une représentation en grandeur réelle du caractère édité, et une fonction d'aide permettant de repérer rapidement toutes les fonctions du menu.
Tous les déplacements du curseur au sein de la grille s'effectuent par l'intermédiaire du pavé numérique, offrant ainsi un déplacement dans 8 directions. La fonction "V" permet d'enchaîner une suite de matrices ou de groupes de matrices reproduisant ainsi de cours dessins animés.
Une ombre vient se glisser au tableau, la lenteur du basic qui vient quelquefois gêner l'utilisateur, en particulier lors de l'affichagee d'une matrice, ou des opérations de codage. Un bon logiciel dans l'ensemble, gratifié d'un démo située après le programme mais sans grande consistance.



Gérer un magasin de chaussure et une fédération sportive.

CPC la bonne pointure

Il a 29 ans. Il est gérant du magasin "La boîte à grolles" et président d'une Fédération Française de Sport: le Racquetball. Il se passionne aussi pour la communication, l'électronique, l'aviation et l'histoire. Etudiant, il a travaillé au rayon Haute Fidélité chez Darty, où sa passion pour ce qu'il vend le place en tête du chiffre d'affaire réalisé par les vendeurs. Il a vécu trois ans au Canada, et c'est de là qu'il a "rapporté" le Racquetball*, sport passionnant, facile d'accès et plaisant à tout niveau. Gilbert Serrat, vous invite à faire comme lui: le pratiquer ! Ces deux activités diverses ont deux points communs: le dynamisme de leur dirigeant, et l'AMSTRAD CPC 464.

En effort, dès janvier dernier, Gilbert décide de s'équiper d'un micro-ordinateur. Après deux mois d'investigation son choix se porte sur l'Amstrad. Son rapport qualité/prix, ses performances, le développement des logiciels, surtout à caractère professionnel, sont les éléments majeurs de cette décision. Un écran monochrome, une imprimante Centronics GLP et un adaptateur péritel pour téléviseur couleur complètent le CPC 464. Cette formule permet une économie de près de mille francs sur l'ensemble comprenant le moniteur couleur.

Pour la "Boîte à grolles", sa boutique de chaussures, Gilbert a recherché parmi les logiciels existants ceux qui pourraient répondre à ses besoins de comptabilité. C'est ainsi que la programme de chez CORE, Multi-gestion, a été éprouvé sur des besoins réels. En fait, comme annoncé par le concepteur, les possibilités d'utilisation sont assez limitées et n'ont permis que les entrées par type de règlements: cartes de crédit, chèques ou espèces, et les sorties dépenses par jour. En fin de jour un total recettes, dépenses est fait, permettant de tirer le solde et d'en faire le cumul par mois puis par année.

Mais attention à ne pas se tromper! Car si vous entrez une somme fausse: impossible de la corriger. Vous êtes obligés de passer la même "écriture" en dépense ou en recette, en fonction de l'erreur commise, afin de la compenser.



L'inconvénient d'un tel système est la création d'écritures qui n'ont pas lieu d'être, au moment de tirer la balance. Il est par ailleurs impossible de revenir au "menu" après avoir entré les données. Impossibilité, également, d'imprimer un mois complet jour par jour... Dommage. C'est des fois, bien utile! En fait, et comme annoncé, ce logiciel permet plus de "taquiner" la gestion familiale que celle plus spécifique de petites sociétés.

"Mon problème a été la correction de sommes où j'avais inversé des chiffres, et aussi d'être obligé de tirer sur listing jour par jour un mois déterminé" affirme Gilbert.

Comptamstrad: bien, mais un peu lent

C'est enfin après quelques mois, que VISMEDIT sort en programme "COMPTAMSTRAD" permettant de tenir une véritable comptabilité: plan comptable, ouverture des comptes: TVA, achats, ventes... Gilbert Serrat voit enfin la possibilité de jeter ses cahiers, et autres livres de comptabilité au rebut pour pouvoir gérer de façon "professionnelle" et à moindre coût sa "Boîte à grolles". Prix d'achat du logiciel 450 F (contre 250 F pour Multi-Gestion). N'oublions pas la TVA récupérable de 18.60% avis aux professionnels !

Notre utilisateur vous conseille, conjointement, l'achat de deux ou trois magazines annexes (Laser Magazine, CB Magazine) afin de sentir le temps de chargement passer plus vite. Il est vrai qu'il lui a fallu beaucoup de patience afin d'arriver à faire tourner ce programme. Et encore beaucoup de patiente pour recharger le programme, lorsque celui-ci s'est planté à peu près une fois sur deux !

Cependant lorsque tout fonctionne, enfin normalement, les possibilités de ce logiciel sont intéressantes. C'est ainsi que Gilbert a ouvert ses comptes, selon le nouveau plan comptable, mais la numérotation des comptes ne se fait qu'à quatre chiffres donc pas de sous-comptes. De plus lorsque vous créez le plan comptable il faut faire bien attention à ce que l'ordre numérique des comptes soit bien tenu, car il n'existe pas de possibilité (ou il ne l'a pas trouvé) de remise en ordre. Ici pas question de se tromper lors de l'entrée des données car, pour corriger il vous faudrait passer une contre écriture, à juste titre puisqu'en comptabilité c'est ce principe qui fait loi.

Dans l'ensemble donc, un logiciel intéressant bien présenté dans un étui en plastique vert et qui, comme pour notre gestionnaire vous apprendra à "gérer votre patience !". Son seul commentaire a été: "c'est lent, très lent".

Quatre cents fiches pour le Racquetball

L'autre activité de Gilbert Serrat c'est le Racquetball. Il vous entend déjà demander : "*Mais qu'est-ce que c'est le* Racquetball ?" Alors patiemment il vous expliquera que: "c'est un sport qui se joue dans un court fermé de 12,20 m de long sur 6,10 m de large et 6,10 m de haut. Toutes les parties du court-plafond et mur arrière compris sont parties intégrantes du jeu. Une raquette à grand tamis et petit manche et une balle qui rebondit à 1,70 m minimun et 2,50 m maximun du sol forment l'équipement. Vingt-cinq millions de joueurs à travers le monde pratiquent ce sport avec plaisir car l'attrait est que l'on joue à son rythme. Nous ne sommes pas tous des champions. Fin juin dernier la France a terminé 5e au championnat d'Europe qui s'est tenu à Anvers en Belgique".

Aussi, dans le cadre de la Fédération: Fédération Française des Associations de Racquetball, il lui faut gérer ses licenciés, et les programmes de gestion de fichiers ne sont pas aussi nombreux que les joueurs de Racquetball. Notre Président s'est tourné, encore une fois, vers CORE qui dispose d'un programme de "Gestion de Fichiers". Malheureusement il n'a eu le plaisir d'avoir, à l'achat la notice d'explication. Aussi, c'est en prenant son courage à deux mains qu'il a découvert ce logiciel.

Seulement deux cents fiches peuvent être gérées, il faut donc prévoir une rechargement de programme pour traiter deux cents nouvelles fiches. Les explications en général sont trop succintes et il faut de la pratique pour bien les saisir. L'ouverture des rubriques est assez facile, mais la saisie des données est très longue à faire. Surtout n'oubliez pas d'indiquer le numéro de la rubrique pour chaque saisie, sinon l'impression se fera sans indication. Le tri a besoin d'être plus explicite, et détaillé. Il permet toutefois par sa recherche alpha-numérique de retrouver certaines fiches.

Tous ces logiciels offrent déjà une bonne approche de l'informatique professionnelle, mais ne peuvent permettre à des machines, tel l'Amstrad, de fournir ce que des responsables en attendent comme proposé dans leurs descriptifs. Aussi Gilbert Serrat va se mettre de façon plus active à la programmation, pour essayer de tirer le maximum de son CPC 464. Ce que je crains pour ce dynamique Président-Gérant, c'est que ses programmes de gestion ne prennent l'allure de programmes de matchs de Racquetball !

Jacques Gilbert

** Fédération Française des Associations de Racquetball, 80, Avenue Jean-Jaurès, 75019 Paris. Tél. (1) 200.95.45 - (1) 671.78.91*



Des tabulations inédites

Le manuel du CPC est très complet et explique très clairement le fonctionnement de l'éditeur. Mais les auteurs ont oublié certaines fonctions de tabulation inexistantes dans le manuel, mais présentes dans la ROM.

Ainsi CTRl "" permet, lors de l'édition d'une ligne de revenir au premier caractère de celle-ci, CTRl "", exécute la fonction inverse, CTRL "'", reporte le curseur au début de la ligne BASIC et CTRL "&" ramène en fin de ligne.

Pour contrôler plus facilement son joystick

Les concepteurs de la ROM du CPC 464, ainsi que ceux du CPC 664 ont eu la bonne idée de ne pas filtrer dans leurs routines de test clavier, en mode édition, les entrées du port joystick.

Ainsi les différentes positions de celui-ci sont considérées comme des touches par l'Amstrad. Ces touches correspondent aux caractères suivants :'&.

Ainsi, si, lors d'un jeu ou d'un programme vous insérez une ligne Basic du type: IF INKEY$ = " ' " THEN...
(le caractère " ' " étant directement entré au joystick) vous obtiendrez un test direct du joytsick.

A propos des caractères de contrôle

Saviez-vous que sur l'Amstrad, il vous est possible de rentrer directement au clavier les différents caractères de contrôles de la norme ASCII ?

Ainsi lorsque vous tapez CTRL accompagné d'une lettre comprise entre A et Z, vous obtiendrez respectivement les codes ASCII compris entre 1 et 26. Exemple: CTRL + "G" correspond au BEl ASCII, c'est-à-dire à la sonnerie.

Il est très important de filtrer ces caractères lors d'un programme utilisant les instructions INKEY$ ou INPUT, car ceux-ci peuvent, par exemple à la suite d'une fausse manœuvre, venir perturber le bon fonctionnement du programme.

Une erreur dans la ROM

Une petite erreur est venue se glisser dans la ROM des lecteurs de disques AMSTRAD. Lors d'une demande de catalogue, si le disque n'a pas été introduit dans le lecteur, l'ordinateur affiche le message: RETRY, IGNORE OR CANCEL.

Si l'on tape, plusieurs fois "IGNORE", l'AMSTRAD, après plusieurs passages éxécute une lecture factice et affiche un contenu du disque aléatoire.

Eric Charton